UNCLASSIFIED

AD NUMBER
ADB015216
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; NOV 1975. Other requests shall be referred to Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH 45433.**
AUTHORITY
AFFDL ltr 2 May 1979

THIS PAGE IS UNCLASSIFIED

AFFDL-TR-75-146

ADB015216

ACTIVE CONTROL SYNTHESIS FOR FLEXIBLE VEHICLES
Volume II KONPACT Program Listing

HONEYWELL
SYSTEMS & RESEARCH CENTER
2600 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

JULY 1976

TECHNICAL REPORT AFFDL-TR-75-146 FOR PERIOD APRIL 1975 - APRIL 1976

Distribution limited to U. S. Government agencies only; test and evaluation statement applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FCG), Wright-Patterson Air Force Base, Ohio 45433.

Prepared for
U.S. AIR FORCE FLIGHT DYNAMICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Distribution limited to U. S. Government agencies only; test and evaluation statements applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FGC), Wright-Patterson Air Force Base, Ohio 45433.

This technical report has been reviewed and is approved for publication.

Charles R. Stockdale
Project Engineer

FOR THE COMMANDER

Evard H. Flinn, Chief
Control Criteria Branch
Air Force Flight Dynamics Laboratory

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE - 28 OCTOBER 76 - 200

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 AFFDL-TR-75-146- Vol-[redacted] 2	2. GOV'T ACCESSION NUMBER	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (AND SUBTITLE) 6 ACTIVE CONTROL SYNTHESIS FOR FLEXIBLE VEHICLES. Volume II. KONPACT Program Listing.		5. TYPE OF REPORT/PERIOD COVERED Final Report, Apr[redacted]975 [redacted] Apr[redacted]976
	14	6. PERFORMING ORG. REPORT NUMBER 76SRC20-Vol-2
7. AUTHOR(S) 10 A. F. Konar, J. K. Mahesh, C. R. Stone, M. Hank	15	8. CONTRACT OR GRANT NUMBER(S) F33615-75-C-3046
9. PERFORMING ORGANIZATIONS NAME/ADDRESS Honeywell Inc., Systems and Research Center 2600 Ridgway Parkway N.E. Minneapolis, Minnesota 55413		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 8219, Task 82190221 16 17
11. CONTROLLING OFFICE NAME/ADDRESS U. S. Air Force Flight Dynamics Laboratory Wright Patterson Air Force Base, Ohio 45433	11	12. REPORT DATE Jan[redacted]1976
	22	13. NUMBER OF PAGES 308 326p.
14. MONITORING AGENCY NAME/ADDRESS (IF DIFFERENT FROM CONT. OFF.)		15. SECURITY CLASSIFICATION (OF THIS REPORT) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (OF THIS REPORT)

Distribution limited to U.S. Government agencies only; test and evaluation statement applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FGC), Wright-Patterson Air Force Base, Ohio 45433.

17. DISTRIBUTION STATEMENT (OF THE ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

Active Control	C-5A	Overlay
CCV (Control Configured Vehicles)	Optimal Control	Variable Dimensioning
Flight	FLEXSTAB	Dynamic Data Storage
Flexible Vehicle	Modeling	PRECOMPILER

20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

KONPACT is a system of computer programs that will design optimal or suboptimal control systems especially for aircraft with lightly damped modes. This program represents advanced computational techniques to perform modern control synthesis, analysis and design of automatic control systems. These programs augment aircraft mathematical models produced from such advanced program as the FLEXSTAB Level 2.01.00 with control system dynamics and then desigr and analyze → next page

HD-168 REV 11/74

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 55 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

402 349

mt

cont.

quadratic optimal or suboptimal control systems.

The KONPACT Program Listings is the second volume of report prepared under contract F33615-75-C-3046.

This report
~~It~~ contains the program listings of KONPACT.

FOREWORD

The research described in this report was prepared by Honeywell Inc., Minneapolis, Minnesota 55413, under Air Force Contract F33615-75-C-3046. It was initiated under the AFFDL task number 82190221, "Optimal Control of Flexible Aircraft," project number 8219 "Stability and Control of Aerospace Vehicles." This work was directed by the Control Criteria Branch (FGC), Flight Control Division of the Air Force Flight Dynamics Laboratory and was administered by Mr. Charles R. Stockdale of the Control Criteria Branch. Special thanks to Mr. Robert C. Schwanz of FGC and Mr. Gary Grimes of ASD/ADDP for their continued support toward this contract.

The technical work reported in this volume was conducted by the Research Department at the Systems and Research Center of Honeywell Inc. Dr. A. F. Konar was the Honeywell Program Manager and the principal investigator on this contract. He was assisted by Mr. C. R. Stone, Dr. J. K. Mahesh, and Miss M. Hank. This report covers work from April 1975 to April 1976.

The work under this contract was reported in three volumes entitled, "Active Control Synthesis for Flexible Vehicles."

Volume I.	KONPACT Theoretical Description	AD-B015 198L
Volume II.	KONPACT Program Listing	
Volume III.	KONPACT Users Manual	AD-B015 025L

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (concluded)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (concluded)

LIST OF TABLES

SECTION I

INTRODUCTION

The general objective of this program is to develop techniques and tools necessary for rapid design of an active control system for aircraft with lightly damped structural modes. The synthesis techniques provided here are aimed at reducing the engineering man-hours presently required for flight control system design thus effecting a cost reduction. Improvements in the fatigue life, ride qualities, and/or handling qualities of military aircraft are sought by controlling the lightly damped modes thus improving mission performance.

The present scope of this program is to develop programs to interface the level 2.01.00 FLEXSTAB computer program system with existing Air Force-owned optimal control computer programs. These programs represent advanced computational techniques required to perform quantitative analysis of multi-surface control systems. The resulting interface program system is called "KONPACT - Computer Programs for Active Control Technology." KONPACT provides the capability to model, synthesize, analyze, and design automatic control systems by efficiently working together with FLEXSTAB. It can also be used as a stand-alone program.

The work performed under this contract is reported in three volumes:

- Volume I. KONPACT Theoretical Description and Demonstration
- Volume II. KONPACT Program Listing
- Volume III. KONPACT Users Manual

This document reports the program listings of KONPACT. Complete documentation of KONPACT is beyond the scope of this contract.

Section II presents a brief description of KONPACT programs. The variable dimensioning technique for efficient data storage and memory allocation is discussed here. This approach is used throughout KONPACT-1.

The Modeling Program (KONPACT-1) is described in Section III. The Design Program (KONPACT-2) is described in Section IV. The appendix contains a description of the precompiler program for KONPACT-1.

The analytical techniques and algorithms used in KONPACT are described in Volume I. Volume I also demonstrates how these techniques are applied to flexible aircraft control system design.

User's information on KONPACT is given in Volume III. The input cards are fully described for each program. Brief descriptions of programs and information flow in KONPACT are also presented for completeness. Demonstration examples are included to guide the user in data mechanics.

SECTION II

DESCRIPTION OF KONPACT PROGRAMS

KONPACT is a system of computer programs developed by Honeywell under Air Force Contract No. F33615-75-C-3046. KONPACT uses the state space approach for modeling flight control systems and designs the controllers using optimal control methodology. KONPACT interfaces with the Linear Systems Analysis (LSA) Program of the Level 2 FLEXSTAB Program system developed by Boeing under Air Force Contract No. F33615-72-C-1172 (Reference 1). KONPACT can also be used as a stand-alone program.

KONPACT operates on CDC6000 and CDC7000 series computers and can be easily modified to operate on other computers. KONPACT has been written in Extended Fortran IV language.

In this section, a description of KONPACT programs is presented in terms of overlay organization and information flow.

OVERLAY ORGANIZATION

KONPACT consists of two programs, namely, a modeling program (KONPACT-1) and a design program (KONPACT-2). KONPACT-1 interfaces with FLEXSTAB through the LSA program to obtain the vehicle model and augments the specified dynamics to obtain the state space description (quadruple data) of the flight control system. These data are utilized by KONPACT-2

which contains the subprograms DIAK and FFOC (documented in Reference 2) to the design of the optimal feedback gains. DIAK stands for Doubly Iterative Algorithm developed by Konar (Reference 5). The DIAK program designs full state feedback optimal controllers. FFOC stands for Fixed Form Optimal Controllers. FFOC stands for Fixed Form Optimal Control developed by Stein and Scharmack (Reference 6). The FFOC program designs reduced state (practical) feedback optimal controllers. KONPACT-2 also interfaces with FLEXSTAB through the LSA program to evaluate performances of the above designed optimal flight control system.

Table 1 provides a brief description of programs KONPACT-1 and KONPACT-2 and their subprograms. The interface between KONPACT and the LSA program is illustrated in Figure 1. The overlay structure of KONPACT-1 program is illustrated in Figure 2. It consists of a main overlay and five primary overlays (Reference 3). The overlay structure of KONPACT-2 program is illustrated in Figure 3. It consists of a main overlay and three primary overlays.

INFORMATION FLOW

The normal sequence for obtaining an overall state space model of a flight control system using the modeling program (KONPACT-1) is as follows:

- The vehicle model is obtained by using either subprogram STAMK1 for LSA data or subprogram STAMK4 for other types of vehicle data.

Table 1. KONPACT Program Descriptions

PROGRAM	SUBPROGRAM	DESCRIPTION
KONPACT-1		State space modeling program
	STAMK1	Obtains state space model from LSA simulator deck data
	STAMK2	Obtains state space model from transfer function data
	STAMK3	Obtains state space model from quadruple data and interconnection data
	STAMK4	Obtains state space model from simulation equations (user written)
	CONDK	Modifies the state space model by scaling, shuffling, truncating and residualizing the system variables
KONPACT-2		Optimal design program
	DATAK	Prepares data for DIAK, FFOC and LSA programs
	DIAK	Designs full state feedback optimal controllers
	FFOC	Designs reduced state (Practical) feedback optimal controllers

- The actuator, sensor, controller, implicit and explicit models are obtained by using either subprogram STAMK2 with transfer function input data or subprogram STAMK3 with quadruple input data.

- The subsystems defined above are combined to get an overall system by using subprogram STAMK3 with interconnection input data.

- The overall system model is conditioned (modified) by scaling and/or shuffling and/or truncating and/or residualizing the variables using the CONDK program. This program also develops the rate of change of response variables when required.

The normal sequence for designing optimal feedback controllers and evaluating the performance of the resulting system using the design program KONPACT-2 is as follows:

- Full state feedback control gains are obtained by varying the quadratic weights and using the DIAK subprogram.

- The resulting full state feedback control gains are reduced to gains only on specified measurements by using the FFOC subprogram.

- The performance of the resulting closed loop system is evaluated using the LSA program.

- The above steps are repeated until a satisfactory design is obtained.

Table 2 describes all the data tapes used in KONPACT-1 and KONPACT-2 programs. The state space model data (quadruple data) and the Name List data are written on tapes QDATA and NDATA, respectively. The

vehicle data (simulator deck data) are written on tape VDATA. The feedback gain data from DIAK and FFOC are written on tapes DDATA and FDATA, respectively. The overall system data in frequency representation form are written on tape SDSTP for use by the LSA program. The DATAK subprogram is used in preparing data tapes for DIAK, FFOC, and LSA.

Table 2. KONPACT Data Tapes

TAPE NAME	DESCRIPTION	GENERATING PROGRAM	BENEFITING PROGRAM(S)
VDATA	Simulator Interface data in the form of card images	LSA	KONPACT-1
QDATA	Quadruple (A, B, C, D) or state variable representation data	KONPACT-1	KONPACT-1 KONPACT-2
NDATA	Name list data of the state variable representation	KONPACT-1	KONPACT-1
DDATA	Full state feedback gain data in the form of card images	KONPACT-2	KONPACT-2
FDATA	Reduced feedback gain data in the form of card images	KONPACT-2	KONPACT-2
SDSTP	Frequency domain representation of quadruple data	KONPACT-2	LSA

VARIABLE DIMENSIONING

Variable dimensioning (dynamic data storage) techniques (Reference 4) are used for efficient data storage. This technique also facilitates changing the amount of allocated (required) storage space by a data card input. In KONPACT the subprogram arrays, whose size depend on the maximum

system dimension inputs, are stored in scratch storage blocks using variable entry points. In the subprograms the arrays are dimensioned with integer variables. These "variable DIMENSION statements" remain unchanged although the amount of required data storage is altered. The maximum size of the scratch storage blocks is specified, in a "fixed DIMENSION statement," in the main program.

The size of storage actually needed by the arrays varies depending on the maximum system dimension inputs. Thus, if the maximum size a user allows for his program changes, there are only the "fixed DIMENSION statements," in the main program, to be changed. Changing the main program of KONPACT-1 is done by a precompiler, as discussed in Section V. The user provides the new maximum system dimensions by data cards. Updating and running with the updated main program are done with control cards in a single run.

In KONPACT programs, four scratch storage blocks, namely S1, S2, S3, and S4 are used. These are specified in the MAIN program of main overlay in labeled COMMON statements under SC1, SC2, SC3, and SC4, respectively. The maximum sizes of these scratch storage blocks are defined there.

The main programs in the primary overlays perform four specific tasks of variable dimensioning. A primary overlay main program first defines the scratch storage blocks under labeled COMMON statements as follows:

```
COMMON/SC1/S1(1)
COMMON/SC2/S2(1)
COMMON/SC3/S3(1)
COMMON/SC4/S4(1)
```

Second, it calculates the start indexes (N1, N2, ... etc.) of the scratch arrays for the stored data as shown in Table 3. Third, it checks the total length occupied by the arrays against the size of the allocated scratch storage blocks. Fourth, it passes the start indexes of the arrays to the subprograms.

Table 3. Typical Dynamic Storage Map

Storage Block	Arrays	Block Addresses
S1 (N1)	V(MAXN)	N1 = 1
S1 (N2)	W(MAXM)	N2 = N1 + MAXN
S1 (N3)	F(MAXN, MAXM)	N3 = N2 + MAXM
S1 (N4)	U(NUM)	N4 = N3 + MAXN * MAXM

SECTION III

MODELING PROGRAM (KONPACT-1)

KONPACT-1 interfaces with FLEXSTAB through the LSA program to obtain the unaugmented vehicle model. It augments this model with the specified dynamics (actuator, sensor, controller, gust, etc.) to obtain the state space description (quadruple data) of the overall flight control system for design.

In this section, a description of the KONPACT-1 program is presented in terms of overlay structure, flow charts, and program listings.

OVERLAY STRUCTURE

The KONPACT-1 program consists of a main overlay and five primary overlays. The overlay structure and the subroutines in each overlay are given in Figure 4. The subroutine summary consisting of name, description, reference, overlay position, and interrelationship is given in Table 4.

DESCRIPTION OF MAIN PROGRAMS

Program MAIN

This is the main program for overlay (0,0). This program assigns the various file numbers used in KONPACT-1. Maximum system dimensions

Table 4. KONPACT-1 Subroutine Summary

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
MAIN	Sets up system dimensions and scratch array dimensions.		5	0,0	KORG1	
KORG1	Organizes input data and calls the primary overlays.	11	12	0,0	FILE IDPR IDRO	MAIN
NAMEL	Reads, prints and updates name list data for the systems.	13	14	0,0	FILE HPR	STAMK1 STAMK2 STAMK3 QUADK STAMK4
QDIO	Reads and writes quadruple data.	57	58	0,0	MPRS FILE	STAMK1 STAMK2 STAMK3 QUADK STAMK4 RESPK
IDRO	Reorganizes the input data.	59	60	0,0		KORG1
FILE	Locates and inserts system labels on disc files and writes end of data mark on the disc files.	65	66	0,0		KORG1 NAMEL QDIO SIMK MNAME
TPR	Prints transfer function data.	67	68	0,0		SIMKT
HPR	Prints heading for the system name.		69	0,0		NAMEL STAMK1 STAMK2 STAMK3 STAMK4 MNAME

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship Calls	Called by
IDPR	Prints input data.		70	0,0		KORG1
MPRS	Prints matrix data on line printer.		71	0,0		QDIO STAMK1 STAMK2 SIMKT STAMK3 SIMK STAMK4 IMRATE REDUCE
ZERO	Initializes (or zeros) the elements of matrices.		73	0,0		QUADK SIMK
INPT	Reads non zero elements of a matrix.		74	0,0		SIMKT QUADK SIMK
DEBUG	Prints a debugging message.		76	0,0		STAMK1 SIMK1 MAIN2 STAMK2 SIMKT DFN PHERR TRANSK MAIN5 RESPK MNAME RSDRD SDRD SHIFT
ERRM	Prints error message.		77	0,0		RESPK MNAME RSDRD SDRD

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
DERRM	Prints an error message when the dimensions for scratch arrays are not sufficient.		78	0,0		MAIN1 STAMK1 SIMK1 MAIN2 STAMK2 MAIN3 STAMK3 MAIN4 STAMK4 MAIN5
DERRMS	Prints an error message when the system dimensions are not sufficient.		79	0,0		STAMK1 STAMK2 STAMK3 STAMK4
TDINVR	Inverts a non-singular matrix or solves a set of linear equations.		81	0,0		STAMK1 STAMK2 STAMK3 STAMK4 IMRATE REDUCL
MAIN1	Sets up block addresses and checks if scratch array size is sufficient.		6	1,0	DERRM STAMK1	
STAMK1	Obtains state space model from LSA simulator deck data and load equation data (implemented in SIMK1 subroutine).	15	16	1,0	SIMK1 DERRM TDINVR MPRS NAMEL DEBUG QDIO HPR	MAIN1
SIMK1	Reads simulator deck data and load equation data and implements them into simulation equations.	17	18	1,0	DEBUG DERRM INPT1 MPRS1	STAMK1
MPRS1	Prints simulator deck data and load equation data.		72	1,0		SIMK1

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
INPT1	Reads simulator deck data and Load equation data.		75	1,0		SIMK1
MAIN2	Sets up block addresses and checks if scratch array size is sufficient.		7	2,0	DERRM DEBUG STAMK2	
STAMK2	Obtains state space model from Transfer function data and connection data (implemented in SIMKT subroutine).	19	20	2,0	SIMKT DERRM TDINVR DEBUG MPRS HPR QDIO DERRMS NAMEL	MAIN2
SIMKT	Reads transfer function data and Connection data and implements them into simulation equations.	21	22	2,0	DEBUG TPR DFN PHERR TRANSK INPT MPRS	STAMK2
TRANSK	Computes state space model for rational transfer functions of up to 5th order.	23	24	2,0	DEBUG	SIMKT
DFN	Selects the specified pade approximation to transport (time) delay from a table of pade approximations.	25	26	2,0	DEBUG	SIMKT
PHERR	Computes the phase error of pade approximation to transport (time) delay.	27	28	2,0	DEBUG	SIMKT
MAIN3	Sets up block addresses and checks if scratch array size is sufficient.		8	3,0	DERRM STAMK3	

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
STAMK3	Obtains state space model from state space data of subsystems and Inter-connection data (implemented in SIMK subroutine).	29	30	3,0	SIMK TDINVR DERRM NAMEL QDIO QUADK HPR DERRMS MPRS	MAIN3
SIMK	Reads state space data of subsystems and interconnection data and implements them into simulation equations.	31	32	3,0	ZERO INPUT MPRS FILE	STAMK3
QUADK	Reads directly the state space data for the system.	33	34	3,0	NAMEL QDIO	STAMK3
MAIN4	Sets up block addresses and checks if scratch array size is sufficient.		9	4,0	DERRM STAMK4	
STAMK4	Obtains state space model for the ALDCS controller (implemented in SIMK2 subroutine).	35	36	4,0	SIMK2 DERRM DERRMS TDINVR HPR MPRS NAMEL QDIO	MAIN4
SIMK2	Reads ALDCS controller gains and switch modes and implements ALDCS controller into simulation equations.	37	38	4,0		STAMK4
MAIN5	Sets up block addresses and checks if scratch array size is sufficient.		10	5,0	DEBUG DERRM CONDK	

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
CONDK	Modifies state space data and name list data according to the design specifications.	39	40	5,0	MNAME QDIO DEBUG SDRD SCAL ERRM DIFFK REDUCE SHUFF RSDRD	MAIN5
MNAME	Reads, modifies, and prints the name list data for a system.	41	42	5,0	ERRM DEBUG SHIFT HPR FILE	CONDK
IMRATE	Obtains the Implicit model error rates and truncates the Implicit model.	43	44	5,0	TDINVR MPRS	CONDK
DIFFK	Differentiates either a specified response or state of a system.	45	46	5,0		CONDK
REDUCE	Residualizes or truncates the state space data of a system.	47	48	5,0	TDINVR MPRS	CONDK
SCAL	Computes scaled state space data.	49	50	5,0		CONDK
SHUFF	Shuffles the states space data and name list data for a system.	51	52	5,0	SHUF1 SHUF2	CONDK
SHUF1	Shuffles the specified rows and columns of a matrix.	53	54	5,0		SHUFF
SHUF2	Shuffles the name list data arrays.	55	56	5,0		SHUFF

Table 4. KONPACT-1 Subroutine Summary (Concluded)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
RSDRD	Reads residualization, truncation, and shuffling data.	61	62	5,0	DEBUG ERRM	CONDK
SDRD	Reads scaling data.	63	64	5,0	DEBUG ERRM	CONDK
SHIFT	Shifts the contents of old name list arrays into new name list arrays.		80	5,0	DEBUG	MNAME

and scratch array dimensions are set in this program. The program calls the organizing subroutine KORG1. The program listing is given in Figure 5.

Program MAIN1

This is the main program for overlay (1,0). This program computes the required scratch array dimensions as explained in Section II, and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK1. The program listing is given in Figure 6. The dynamic storage map is given in Table 5.

Program MAIN2

This is the main program for overlay (2,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK2. The program listing is given in Figure 7. The dynamic storage map is given in Table 6.

Program MAIN3

This is the main program for overlay (3,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK3. The program listing is given in Figure 8. The dynamic storage map is given in Table 7.

Table 5. Dynamic Storage Map for Program MAIN1

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORG1
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM*2 + NYM + NUM	Calculated in KORG1
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN*MAXM	U(NUM)	NUM	Defined in MAIN
S2(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S2(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)	NRM	Defined in MAIN
S2(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM*2	DES(NXM, 10)		
S3(L4)	L4 = L3 + NXM*10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM*4	NNO(NRM)		
S3(L6)	L6 = L5 + NRM	VNO(NRM, 2)		
S3(L7)	L7 = L6 + NRM*2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM*10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM*4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM*2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM*10	UNITI(NUM, 4)		

Table 6. Dynamic Storage Map for Program MAIN2

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORG1
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM*2 + NYM + NUM	Calculated in KORG1
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN + MAXM	XDOT(MST, MTFB)	MST	Defined in KORG1
S1(N5)	N5 = N4 + MST*MTFB	X(MST, MTFB)	MTFB	Defined in MAIN
S1(N6)	N6 = N5 + MST*MTFB	BI(MTFB)		
S1(N7)	N7 = N6 + MTFB	UI(MTFB)		
S1(N8)	N8 = N7 + MTFB	U(NUM)	NUM	Defined in MAIN
S1(N9)	N9 = N8 + NUM	NNX(MTFB)		
S1(N10)	N10 = N9 + MTFB	NNR(MTFB)		
S1(N11)	N11 = N10 + MTFB	NNU(MTFB)		
S2(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S2(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)	NRM	Defined in MAIN
S2(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM*2	DESS(NXM, 10)		
S3(L4)	L4 = L3 + NXM*10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM*4	NNO(NRM)		
S3(L6)	L6 = L5 + NRM	VNO(NRM, 2)		
S3(L7)	L7 = L6 + NRM*2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM*10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM*4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM*2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM*10	UNITI(NUM, 4)		

Table 7. Dynamic Storage Map for Program MAIN3

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORG1
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM*2 + NYM + NUM	Calculated in KORG1
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN*MAXM	XDOT(NXM, MB)	NXM	Defined in MAIN
S1(N5)	N5 = N4 + NXM*MB	X(NXM, MB)	MB	Defined in MAIN
S1(N6)	N6 = N5 + NXM*MB	RI(NRM, MB)	NRM	Defined in MAIN
S1(N7)	N7 = N6 + NRM*MB	UI(NUM, MB)	NUM	Defined in MAIN
S1(N8)	N8 = N7 + NUM*MB	U(NUM)		
S1(N9)	N9 = N8 + NUM	RIN(NRMMB)	NRMMB = NRM*MB	Calculated in MAIN3
S1(N10)	N10 = N9 + NRMMB	NNX(MB)		
S1(N11)	N11 = N10 + MB	NNR(MB)		
S1(N12)	N12 = N11 + MB	NNU(MB)		
S2(M1)	M1 = 1	A(NXM, NXM)		
S2(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM*2	DESS(NXM, 10)		
S3(L4)	L4 = L3 + NXM*10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM*4	NNO(NRM)		
S3(L6)	L6 = L5 + NRM	VNO(NRM, 2)		
S3(L7)	L7 = L6 + NRM*2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM*10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM*4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM*2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM*10	UNITI(NUM, 4)		

Program MAIN4

This is the main program for overlay (4, 0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK4. The program listing is given in Figure 9. The dynamic storage map is given in Table 8.

Program MAIN5

This is the main program for overlay (5, 0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the conditioning subroutine CONDK. The program listing is given in Figure 10. The dynamic storage map is given in Table 9.

DESCRIPTION OF BASIC SUBROUTINES

Subroutine KORG1

This subroutine organizes the execution of KONPACT-1 program. The input data cards for KONPACT-1 program are read and printed by this subroutine. The print specification cards are read in this subroutine and the print control parameter IPRINT is set for the printer output options of KONPACT-1 program. The flow chart is given in Figure 11 and the program listing is given in Figure 12.

Table 8. Dynamic Storage Map for Program MAIN4

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORG1
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM*2 + NYM + NUM	Calculated in KORG1
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN*MAXM	U(NUM)	NUM	Defined in MAIN
S2(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S2(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)	NRM	Defined in MAIN
S2(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM*2	DESS(NXM, 10)		
S3(L4)	L4 = L3 + NXM*10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM*4	NNO(NRM)		
S3(L6)	L6 = N5 + NRM	VNO(NRM, 2)		
S3(L7)	L7, L6 + NRM*2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM*10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM*4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM*2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM*10	UNITI(NUM, 4)		

Table 9. Dynamic Storage Map for Program MAIN5

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(L1)	L1 = 1	DUMMY1(NDM11, NDM12)	NDM11 = MAX(17, NXM, NRM)	Calculated in MAIN5
S1(L2)	L2 = L1 + NDM11*NDM12	DUMMY2(NDM21, NDM22)	NDM12 = MAX(NXM + NUM, NRM)	Calculated in MAIN5
S1(L3)	L3 = L2 + NDM21*NDM22	DUMMY3(NUM)	NDM21 = MAX(NRM, NXM)	Calculated in MAIN5
S1(L4)	L4 = L3 + NUM	ES(NXM, NUM)	NDM22 = MAX(NXM, NRM, NUM)	Calculated in MAIN5
S1(L5)	L5 = L4 + NXM*NUM	ER(NRM, NUM)	NRM	Defined in MAIN
S1(L6)	L6 = L5 + NRM*NUM	NSHUFS(NXM)	NXM	Defined in MAIN
S1(L7)	L7 = L6 + NXM	NSHUFO(NRM)		
S1(L8)	L8 = L7 + NRM	NSHUFI(NUM)	NUM	Defined in MAIN
S1(L9)	L9 = L8 + NUM	CS(NRM, NXM)		
S1(L10)	L10 = L9 + NRM*NXM	DS(NRM, NUM)		
S1(L11)	L11 = L10 + NRM*NUM	CW(NRM, NXM)		
S1(L12)	L12 = L11 + NRM*NXM	DW(NRM, NUM)		
S1(L13)	L13 = L12 + NRM*NUM	IRS(NRM)		
S1(L14)	L14 = L13 + NRM	Q(NRM, NRM)		
S2(M1)	M1 = 1	A(NXM, NXM)		
S2(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S2(M5)	M5 = M4 + NRM*NUM	CM(NRM, NXM)		
S2(M6)	M6 = M5 + NRM*NXM	DM(NRM, NUM)		
S3(N1)	N1 = 1	NNS(NXM)		
S3(N2)	N2 = N1 + NXM	VNS(NXM, 2)		
S3(N3)	N3 = N2 + NXM*2	DESS(NXM, 10)		
S3(N4)	N4 = N3 + NXM*10	UNITS(NXM, 4)		

Table 9. Dynamic Storage Map for Program MAIN5 (Concluded)

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S3(N5)	N5 = N4 + NXM*4	NNO(NRM)		
S3(N6)	N6 = N5 + NRM	VNO(NRM, 2)		
S3(N7)	N7 = N6 + NRM*2	DESO(NRM, 10)		
S3(N8)	N8 = N7 + NRM*10	UNITO(NRM, 4)		
S3(N9)	N9 = N8 + NRM*4	NNI(NUM)		
S3(N10)	N10 = N9 + NUM	VNI(NUM, 2)		
S3(N11)	N11 = N10 + NUM*2	DESI(NUM, 10)		
S3(N12)	N12 = N11 + NUM*10	UNITI(NUM, 4)		
S3(N13)	N13 = N12 + NUM*4	NNNS(NXM)		
S3(N14)	N14 = N13 + NXM	VNNS(NXM, 2)		
S3(N15)	N15 = N14 + NXM*2	DESNS(NXM, 10)		
S3(N16)	N16 = N15 + NXM*10	UNITNS(NXM, 4)		
S3(N17)	N17 = N16 + NXM*4	NNNO(NRM)		
S3(N18)	N18 = N17 + NRM	VNNO(NRM, 2)		
S3(N19)	N19 = N18 + NRM*2	DESNO(NRM, 10)		
S3(N20)	N20 = N19 + NRM*10	UNITNO(NRM, 4)		
S3(N21)	N21 = N20 + NRM*4	NNNI(NUM)		
S3(N22)	N22 = N21 + NUM	VNNI(NUM, 2)		
S3(N23)	N23 = N22 + NUM*2	DESNI(NUM, 10)		
S3(N24)	N24 = N23 + NUM*10	UNITNI(NUM, 4)		

Subroutine NAMEL

This subroutine obtains the name list data for the system variables. The subroutine either reads the name list data from input data cards or internally obtains a default name list data. In the case of combining various subsystems into an overall system, the subroutine uses the interconnection data to obtain the appropriate name list data. This subroutine also writes the name list data of each system on NDATA file for use by other programs. The flow chart is given in Figure 13 and the program listing is given in Figure 14.

Subroutine STAMK1

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK1. The flow chart is given in Figure 15 and the program listing is given in Figure 16. The dynamic storage map is given in Table 10.

Subroutine SIMK1

This subroutine reads simulator deck data and load equation data obtained by the Linear System Analysis (LSA) program and implements them into simulation equations. The flow chart is given in Figure 17 and the program listing is given in Figure 18.

Subroutine STAMK2

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMKT. The flow chart is given in

Table 10. Dynamic Storage Map for Program STAMK1

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
W(N1)	N1 = 1	XDOT(NX)	NX	Calculated in SIMK1
W(N2)	N2 = N1 + NX	Y(NY)	NY	Calculated in SIMK1
W(N3)	N3 = N2 + NY	X(NX)		
W(N4)	N4 = N3 + NX	U(NU)	NU	Calculated in SIMK1
V(N1)	N1 = 1	XDOTL(NX)		
V(N2)	N2 = N1 + NX	YL(NY)		
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMK1
S1(L1)	L1 = 1	DESSS(NXM, 10, MB)	NXM,MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNITOO(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESII(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L9)	L9 = L8 + MB	NUU(MB)		

Figure 19 and the program listing is given in Figure 20. The dynamic storage map is given in Table 11.

Subroutine SIMKT

This subroutine reads transfer function data and connection data and implements them into simulation equations. The flow chart is given in Figure 21 and the program listing is given in Figure 22.

Subroutine TRANSK

This subroutine computes the state space model for rational transfer functions using the input Frobenius form of realization. The flow chart is given in Figure 23 and the program listing is given in Figure 24.

Subroutine DFN

This subroutine selects the specified Pade approximation to transport (time) delay from a table of Pade approximations. The flow chart is given in Figure 25 and the program listing is given in Figure 26.

Subroutine PHERR

This subroutine computes the phase error of the Pade approximation to transport (time) delay. The flow chart is given in Figure 27 and the program listing is given in Figure 28.

Table 11. Dynamic Storage Map for Program STAMK2

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
V(N1)	N1 = 1	XDOL(NX)	NX	Calculated in SIMKT
V(N2)	N2 = N1 + NX	YL(NY)	NY	Calculated in SIMKT
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMKT
S1(L1)	L1 = 1	DESSS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNITOO(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESII(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L7)	L9 = L8 + MB	NUU(MB)		

Subroutine STAMK3

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK. The flow chart is given in Figure 29 and the program listing is given in Figure 30. The dynamic storage map is given in Table 12.

Subroutine SIMK

This subroutine reads interconnection data and state space data for subsystems and implements the simulation equations for the overall system. SIMK also writes the interconnection data on the scratch file for use by subroutine NAMEL. The flow chart is given in Figure 31 and the program listing is given in Figure 32.

Subroutine QUADK

This subroutine reads directly the state space data for the system. The flow chart is given in Figure 33 and the program listing is given in Figure 34.

Subroutine STAMK4

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK2. The flow chart is given in Figure 35 and the program listing is given in Figure 36. The dynamic storage map is given in Table 13.

Table 12. Dynamic Storage Map for Program STAMK3

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
V(N1)	N1 = 1	XDOTL(NX)	NX	Calculated in SIMK
V(N2)	N2 = N1 + NX	YL(NY)	NY	Calculated in SIMK
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMK
S1(L1)	L1 = 1	DESSS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNIT))(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESII(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L9)	L9 = L8 + MB	NUU(MB)		
S2(M1)	M1 = 1	AT(NXM, NXM, MB)		
S2(M2)	M2 = M1 + NXM*NXM*MB	BT(NXM, NUM, MB)		
S2(M3)	M3 = M2 + NXM*NUM*MB	CT(NRM, NXM, MB)		
S2(M4)	M4 = M3 + NRM*NXM*MB	DT(NRM, NUM, MB)		
S2(M5)	M5 = M4 + NRM*NUM*MB	P(MN, MN)	MN=MM*MB	Calculated in KORG1

Table 12. Dynamic Storage Map for Program STAMK3 (Concluded)

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S2(M6)	M6 = M5 + MN*MN	Q(MN, NUM)		
S2(M7)	M7 = M6 + MN*NUM	R(NRM, MN)		
S2(M8)	M8 = M7 + NRM*MN	S(NRM, NUM)		
S3(K1)	K1 = 1	PP(MP, MM, MM)	MN = MAX (NRM, NUM)	Calculated in KORG1
S3(K2)	K2 = K1 + MP*MM*MM	QQ(MQ, MM, NUM)	MQ = MB	Calculated in KORG1
S3(K3)	K3 = K2 + MQ*MM*NUM	RR(MR, NRM, MM)	MR = MB	Calculated in KORG1
S3(K4)	K4 = K3 + MR*NRM*MM	NSP(MP)	MP = MB*2	
S3(K5)	K5 = K4 + MP	NSQ(MQ)		
S3(K6)	K6 = K5 + MQ	NSR(MR)		

Table 13. Dynamic Storage Map for Program STAMK4

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
W(N1)	N1 = 1	XDOT(NX)	NX	Calculated in SIMK2
W(N2)	N2 = N1 + NX	Y(NY)	NY	Calculated in SIMK2
W(N3)	N3 = N2 + NY	X(NX)		
W(N4)	N4 = N3 + NX	U(NU)	NU	Calculated in SIMK2
V(N1)	N1 = 1	XDOTL(NX)		
V(N2)	N2 = N1 + NX	YL(NY)		
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMK2
S1(L1)	L1 = 1	DESSS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNITOO(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESII(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L9)	L9 = L8 + MB	NUU(MB)		

Subroutine SIMK2

This is a users written subroutine. Here it is written for the ALDCS controller. It reads ALDCS controller gains and switch modes (positions) and implements the controller into simulation equations. The flow chart is given in Figure 37 and the program listing is given in Figure 38.

Subroutine CONDK

This subroutine organizes the modification (conditioning) of the state space data and name list data according to specification. The flow chart is given in Figure 39 and the program listing is given in Figure 40.

Subroutine MNAME

This subroutine modifies the name list data of the system according to the conditioning data. The flow chart is given in Figure 41 and the program listing is given in Figure 42.

Subroutine IMRATE

This subroutine obtains the implicit model error rates and truncates the implicit model. The flow chart is given in Figure 43 and the program listing is given in Figure 44.

Subroutine DIFFK

This subroutine obtains the rate of change of either a specified response or state of the system by differentiation. If the differentiation requires external rate inputs in the model, a message is printed by the subroutine. The flow chart is given in Figure 45 and the program listing is given in Figure 46.

Subroutine REDUCE

This subroutine residualizes or truncates the state space data of the system. In addition it computes the error of residualization. The flow chart is given in Figure 47 and the program listing is given in Figure 48.

Subroutine SCAL

This subroutine computes the scaled state space data. The flow chart is given in Figure 49 and the program listing is given in Figure 50.

Subroutine SHUFF

This subroutine shuffles the state space data and the name list data by calling subroutines SHUF1 and SHUF2. The flow chart is given in Figure 51 and the program listing is given in Figure 52.

Subroutine SHUF1

This subroutine shuffles the rows and columns of a matrix. The flow chart is given in Figure 53 and the program listing is given in Figure 54.

Subroutine SHUF2

This subroutine shuffles the name list data arrays. The flow chart is given in Figure 55 and the program listing is given in Figure 56.

DESCRIPTION OF AUXILIARY SUBROUTINES

Subroutine QDIO

This subroutine reads the state space data from file QDATA and prints it. It also writes the state space data on file QDATA. The flow chart is given in Figure 57 and the program listing is given in Figure 58.

Subroutine IDRO

This subroutine reorganizes the input data. The reorganized input data is written on file BINPUT. The flow chart is given in Figure 59 and the program listing is given in Figure 60.

Subroutine RSDRD

This subroutine reads residualization, truncation, and shuffling data for the variables of the system. The flow chart is given in Figure 61 and the program listing is given in Figure 62.

Subroutine SDRD

This subroutine reads the scaling factor and the new units for the system variables. The flow chart is given in Figure 63 and the program listing is given in Figure 64.

Subroutine FILE

This subroutine positions the data file for reading or writing data. There are three modes of calling this subroutine. INSERT mode inserts the label name and positions the data file for writing. LOCATE mode locates the label name and positions the data file for reading. NULL mode removes the label name from the data file. The flow chart is given in Figure 65 and the program listing is given in Figure 66.

Subroutine TPR

This subroutine prints transfer function data. The flow chart is given in Figure 67 and the program listing is given in Figure 68.

Subroutine HPR

This subroutine prints the headings for the system label name. The program listing is given in Figure 69.

Subroutine IDPR

This subroutines prints the input data. The program listing is given in Figure 70.

Subroutine MPRS

This subroutine prints a matrix, identifying the rows and columns. The program listing is given in Figure 71.

Subroutine MPRS1

This subroutine prints the simulator interface matrix data from the Linear System Analysis (LSA) program. The program listing is given in Figure 72.

Subroutine ZERO

This subroutine initializes (or zeros) the elements of a matrix. The program listing is given in Figure 73.

Subroutine INPT

This subroutine reads the nonzero elements of a matrix. The program listing is given in Figure 74.

Subroutine INPT1

This subroutine reads the simulator interface matrix data from Linear System Analysis (LSA) program. The program listing is given in Figure 75.

Subroutine DEBUG

This subroutine prints a debugging message. The program listing is given in Figure 76.

Subroutine ERRM

This subroutine prints an error message indicating the program and overlay at which the error was detected. The program listing is given in Figure 77.

Subroutine DERRM

This subroutine prints a message when the maximum dimensions for scratch arrays are not sufficient. The program listing is given in Figure 78.

Subroutine DERRMS

This subroutine prints a message when the Maximum System dimensions are not sufficient. The program listing is given in Figure 79.

Subroutine SHIFT

This subroutine shifts the contents of old name list arrays into new name list arrays. The program listing is given in Figure 80.

Subroutine TDINVR

This subroutine inverts a non-singular matrix or solves a set of linear equations. The program listing is given in Figure 81.

SECTION IV

DESIGN PROGRAM (KONPACT-2)

The data produced by KONPACT-1 are utilized by KONPACT-2. KONPACT-2 contains two Air Force-owned synthesis programs, DIAK and FFOC. The DIAK (Doubly Iterative Algorithm developed by Konar) program computes optimal controller gains for full state feedback. FFOC (Fixed Form Optimal Control) simplifies these gains to specified measurements. KONPACT-2 interfaces with FLEXSTAB through the LSA program to evaluate performances of the closed loop system.

In this section, a description of KONPACT-2 program is presented in terms of overlay structure, flow charts, and program listings. The DIAK and FFOC programs are fully documented in Reference 2 and only the program listings are given here for completeness. Modularization and variable dimensioning of DIAK and FFOC programs are beyond the scope of this contract.

OVERLAY STRUCTURE

The KONPACT-2 program consists of a main overlay and three primary overlays. The overlay structure and the subroutines in each overlay is given in Figure 82. The subroutine summary consisting of name, description, reference, overlay position, and interrelationship is given in Table 14.

Table 14. KONPACT-2 Subroutine Summary

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
MAIN	Sets up system dimensions and scratch array dimensions.		83	0, 0	KORG2	
KORG2	Organizes input data and calls the primary overlays.	87	88	0, 0	IDRO IDPR ERRM	MAIN
IDRO	Reorganizes the input data.			010		KORG2
IDPR	Prints input data.			0, 0		KORG2
ERRM	Prints error message.			0, 0		KORG2 DATAK
TDINVR	Inverts a nonsingular matrix or solves a set of linear equations.			0, 0		DIAK CAL1 FFOC GCAL CAL
MP	Prints matrix data.		110	0, 0		DIAK STRIC RESP FFOC
OUTP	Writes nonzero elements of a matrix on a data file.		111	0, 0		DIAK FFOC
INPT	Reads nonzero elements of a matrix.			0, 0		DIAK FFOC
ZERO	Initializes (or zeros) the elements of a matrix.			0, 0		FFOC DDIAK DFFOC DLSA

Table 14. KONPACT-2 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
POLES	Computes the eignnvalues of a matrix.		112	0,0	HESSEN QRCALL	DIAK FFOC
HESSEN	Reduces a matrix to upper Hessenberg form by Gaussian elimination.		113	0,0		POLES
QRCALL	Computes eigenvalues of an upper Hessenberg matrix.		114	0,0	QR	POLES
QR	Performs a double QR iteration on a real matrix.		115	0,0		QRCALL
DIAK	Computes optimal state feedback gains for a linear time-invariant system with a quadratic cost function.		84	1,0	INPT SHUFL MP STRIC TDINVR CAL1 OUTP TIMER POLES	
TIMER	Computes time response.		89	1,0	SGUST	DIAK
SGUST	Computes step gust input.		90	1,0		TIMER
CAL1	Solves square Lyapunov equation.		91	1,0	TDINVR	COVAR COSTVT DIAK
STRIC	Computes stable set of starting gains for DIAK.		92	1,0	MP	DIAK
SHUFL	Reorders columns and rows of a matrix.		93	1,0		DIAK
GRAN	Generates random numbers.		118	1,0		TIMER

Table 14. KONPACT-2 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship: Calls	Inter-relationship: Called by
FFOC	Computes simplified controller gains for a linear time-invariant system with a quadratic cost function.		85	2,0	POLES OUTP COVAR TRANS COSTAT RESP UNSCR MP ZERO INPT SHUF TDINVR	
SHUF	Reorders rows and columns of matrices.		94	2,0		FFOC
RESP	Computes covariances for disturbance inputs.		95	2,0		FFOC
COVAR	Computes covariance matrix.		96	2,0	CAL GCAL	RESP
COSTAT	Computes costate matrix.		97	2,0	CAL GCAL	FFOC
TRANS	Computes gradient transformation matrix.		98	2,0		FFOC
UNSCR	Transforms the gradient transformation matrix.		99	2,0		FFOC
GCAL	Solves rectangular Lyapunov equation.		100	2,0	TDINVR	COVAR COSTAT
CAL	Solves square Lyapunov equation		101	2,0	TDINVR	COVAR COSTAT
DATAK	Sets up array start indices and checks if scratch array size is sufficient.		86	3,0	DDIAK DFFOC DLSA FINK DERRM ERRM	

Table 14. KONPACT-2 Subroutine Summary (Concluded)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
DDIAK	Prepare data file for DIAK program.	102	103	3,0	ZERO FILE MPRS WTP	DATAK
DFFOC	Prepares data file for FFOC program.	104	105	3,0	ZERO FILE MPRS WTP	DATAK
DLSA	Prepares data for FINK program.	106	107	3,0	ZERO FILE MPRS INPTM	DATAK
FINK	Converts state space data into frequency domain data for LSA program.	108	109	3,0	MPRS	DATAK
MPRS	Prints matrix data.			3,0		DDIAK DFFOC DLSA FINK
FILE	Locates and inserts system labels on disc files and writes end of data mark.			3,0		DDIAK DFFOC DLSA
INPTM	Reads nonzero elements of a matrix.		116	3,0		DDIAK DFFOC DLSA
WTP	Writes nonzero elements of a matrix on a data file.		117	3,0		DDIAK DFFOC
DERRM	Prints an error message when the dimensions for scratch arrays are not sufficient.			3,0		DATAK

DESCRIPTION OF MAIN PROGRAMS

Program MAIN

This is the main program for overlay (0, 0). This program assigns the various file numbers used in KONPACT-2. Maximum system dimensions and scratch array dimensions are set in this program. (Note that scratch arrays should be defined in DATAK program.) The program calls the organizing subroutine KORG2. The program listing is given in Figure 83.

Program DIAK

This is the main program for overlay (1, 0). This program computes optimal state feedback gains for a linear time-invariant system with quadratic cost function. The program listing is given in Figure 84.

Program FFOC

This is the main program for overlay (2, 0). This program computes simplified controller gains for a linear time-invariant system with a quadratic cost function. The program listing is given in Figure 85.

Program DATAK

This is the main program for overlay (3, 0). The scratch arrays are defined here. The program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the data preparation subroutines DDIAK, DFFOC, DLSA and FINK. The program listing is given in Figure 86. The dynamic storage map is given in Table 15.

Table 15. Dynamic Storage Map for Program DATAK

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S1(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)	NUM	Defined in MAIN
S1(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)	NRM	Defined in MAIN
S1(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S2(N1)	N1 = 1	B1(NXM, NUM)		
S2(N2)	N2 = N1 + NXM*NUM	B2(NXM, NUM)		
S2(N3)	N3 = N2 + NXM*NUM	C1(NRM, NXM)		
S2(N4)	N4 = N3 + NRM*NXM	C3(NRM, NXM)		
S2(N5)	N5 = N4 + NRM*NXM	D11(NRM, NUM)		
S2(N6)	N6 = N5 + NRM*NUM	BK(NUM, NRM)		
S2(N7)	N7 = N6 + NUM*NRM	BKC3(NUM, NXM)		
S2(K1)	K1 = 1	CC(NXRM, NXRUM)	NXRM = NXM + NRM	Calculated in DATAK
S2(K2)	K2 = K1 + NXRM*NXRUM	NAME(NXRUM)	NXRUM = NXRM + NUM	Calculated in DATAK

DESCRIPTION OF BASIC SUBROUTINES

Subroutine KORG2

This subroutine organizes the execution of KONPACT-2 program. The input data cards for KONPACT-2 program are read and printed by the subroutine. The print specification cards are read in this subroutine, and the print control parameter IPRINT is set for the printed output options of the KONPACT-2 program. Under the control of the input data this subroutine calls the overlay loader to load the required primary overlay into central memory for execution. The flow chart is given in Figure 87 and the program listing is given in Figure 88.

Subroutine TIMER

This subroutine computes the time response for step command inputs and step gust inputs. The program listing is given in Figure 89.

Subroutine SGUST

This subroutine computes step gust input. The program listing is given in Figure 90.

Subroutine CAL1

This subroutine solves square Lyapunov equation. The program listing is given in Figure 91.

Subroutine STRIC

This subroutine computes a stable set of starting gains for DIAK. The program listing is given in Figure 92.

Subroutine SHUFL

This subroutine reorders the columns and rows of a matrix. The program listing is given in Figure 93.

Subroutine SHUF

This subroutine recordes rows and columns of matrices. The program listing is given in Figure 94.

Subroutine RESP

This subroutine computes covariances for disturbance inputs. The program listing is given in Figure 95.

Subroutine COVAR

This subroutine computes the covariance matrix. The program listing is given in Figure 96.

Subroutine COSTAT

This subroutine computes the costate matrix. The program listing is given in Figure 97.

Subroutine TRANS

This subroutine computes the gradient transformation matrix. The program listing is given in Figure 98.

Subroutine UNSCR

This subroutine transforms the gradient transformation matrix. The program listing is given in Figure 99.

Subroutine GCAL

This subroutine solves the rectangular Lyapunov equation. The program listing is given in Figure 100.

Subroutine CAL

This subroutine solves the square Lyapunov equation. The program listing is given in Figure 101.

Subroutine DDIAK

This subroutine reads data from cards or from file QDATA and prepares data file SCRTCH for DIAK subprogram. The flow chart is given in Figure 102 and the program listing is given in Figure 103.

Subroutine DFFOC

This subroutine reads data from cards or from file QDATA and prepares data file SCRTCH for FFOC subprogram. The flow chart is given in Figure 104 and the program listing is given in Figure 105.

Subroutine DLSA

This subroutine reads data from files QDATA, DDATA, and FDATA and prepares open loop or closed loop state space data. The flow chart is given in Figure 106, and the program listing is given in Figure 107.

Subroutine FINK

This subroutine uses the state space data obtained by the DLSA subroutine, computes the frequency domain data, and writes it on file SDSTP for the LSA program. The flow chart is given in Figure 108, and the program listing is given in Figure 109.

DESCRIPTION OF AUXILIARY SUBROUTINES

Subroutine MP

This subroutine prints matrix data. The program listing is given in Figure 110.

Subroutine OUTP

This subroutine writes the nonzero elements of a matrix on a data file. The program listing is given in Figure 111.

Subroutine POLES

This subroutine computes the eigenvalues of a matrix. The program listing is given in Figure 112.

Subroutine HESSEN

This subroutine computes the upper Hessenberg form of a matrix by Gaussian elimination. The program listing is given in Figure 113.

Subroutine QRCALL

This subroutine computes the eigenvalues of an upper Hessenberg form matrix. The program listing is given in Figure 114.

Subroutine QR

This subroutine performs a double QR iteration on a real matrix. The program listing is given in Figure 115.

Subroutine INPTM

This subroutine reads nonzero elements of a matrix The program listing is given in Figure 116.

Subroutine WTP

This subroutine writes the nonzero elements of a matrix on a data file. The program listing is given in Figure 117.

Function GRAN

This function subroutine generates random numbers. The program listing is given in Figure 118.

For documentation of subroutines IDRO, IDPR, ERRM, TDINVR, INPT, ZERO, MPRS, FILE, and DERRM the reader is referred to Section III.

SECTION V

CONCLUSIONS AND RECOMMENDATIONS

The scope of this program was to develop programs to interface the level 2.01.00 FLEXSTAB with DIAK/FFOC optimal control programs. The theory and algorithms for the interface are presented in Volume I. Two demonstration examples are given in Volume III to show the data mechanics of the interface. A brief documentation of the interface program KONPACT is provided in this volume.

RECOMMENDATIONS FOR FUTURE SOFTWARE DEVELOPMENT WORK

- Full documentation of KONPACT should be made
- DIAK/FFOC programs should be modularized and variable dimensioned
- Faster algorithms should be used to reduce design time
- Reduced Controller Software (FFOC) should be augmented with the minimal order observer design capability

CONCLUSIONS

A large-scale software - KONPACT - for the design and analysis of active control systems is briefly documented in this volume. The work reported in Volumes I, II and III established the total dynamic system approach for the design and analysis.

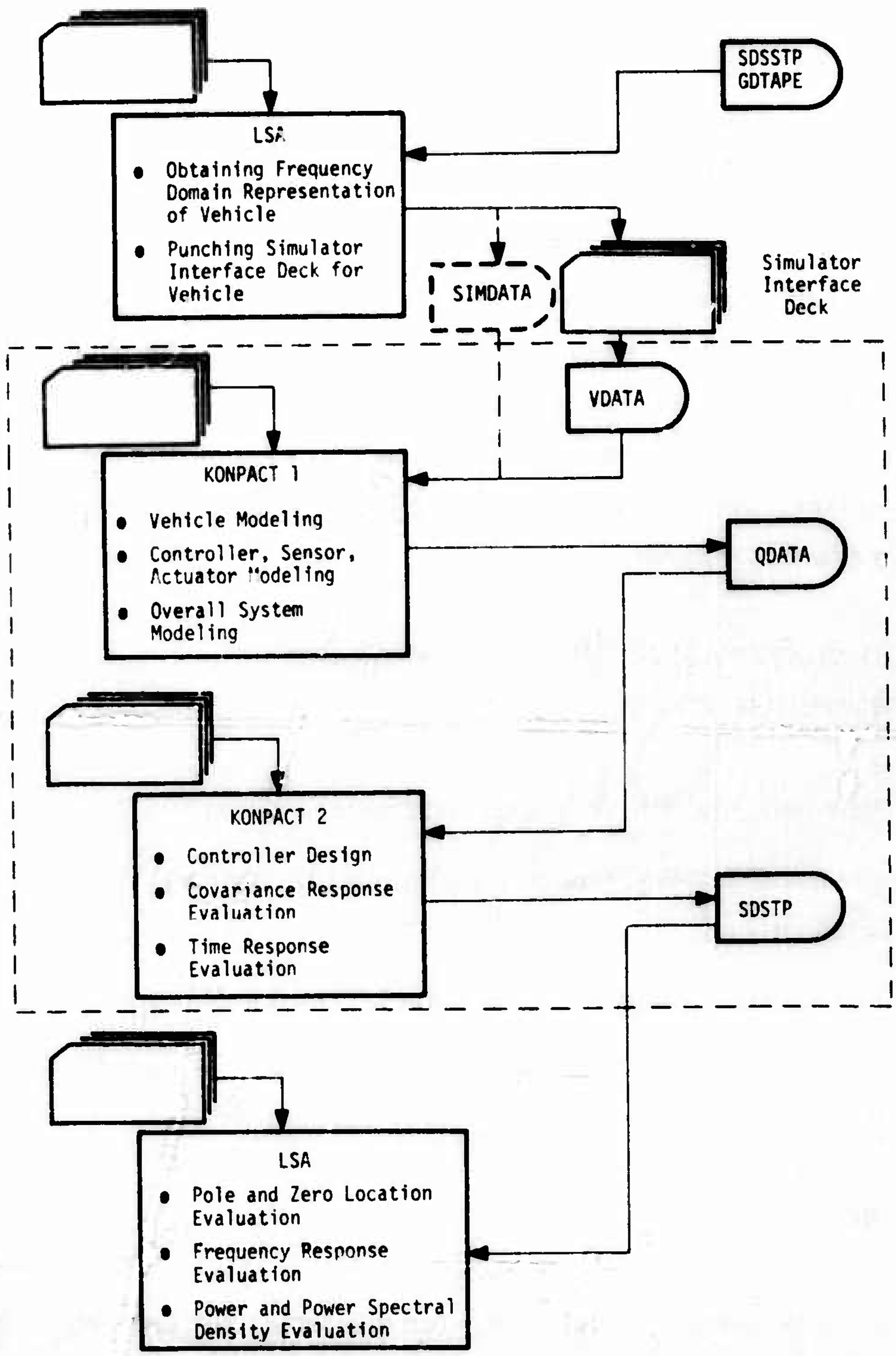

Figure 1. Interface Between LSA and KONPACT Programs

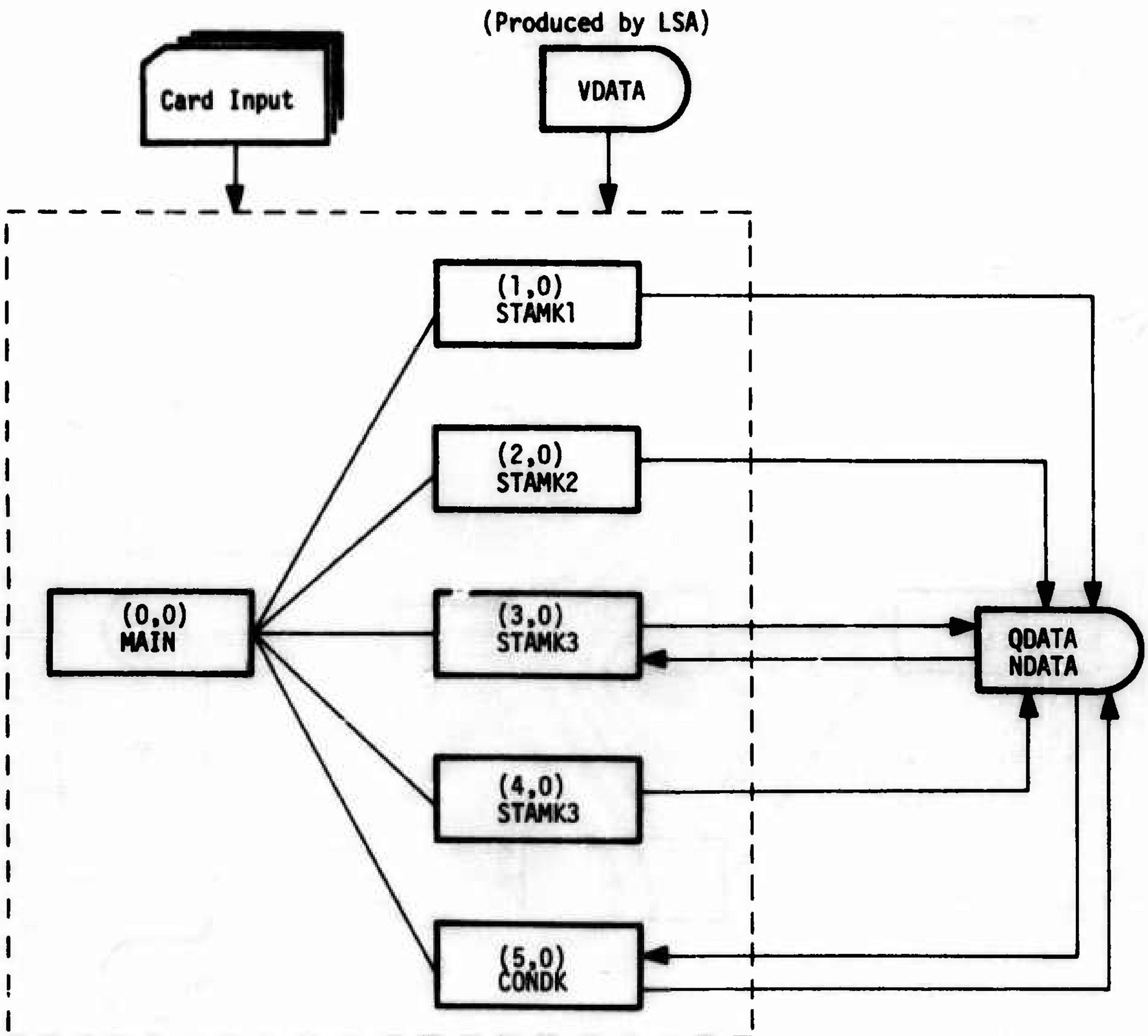

Figure 2. Overlay Structure of KONPACT-1

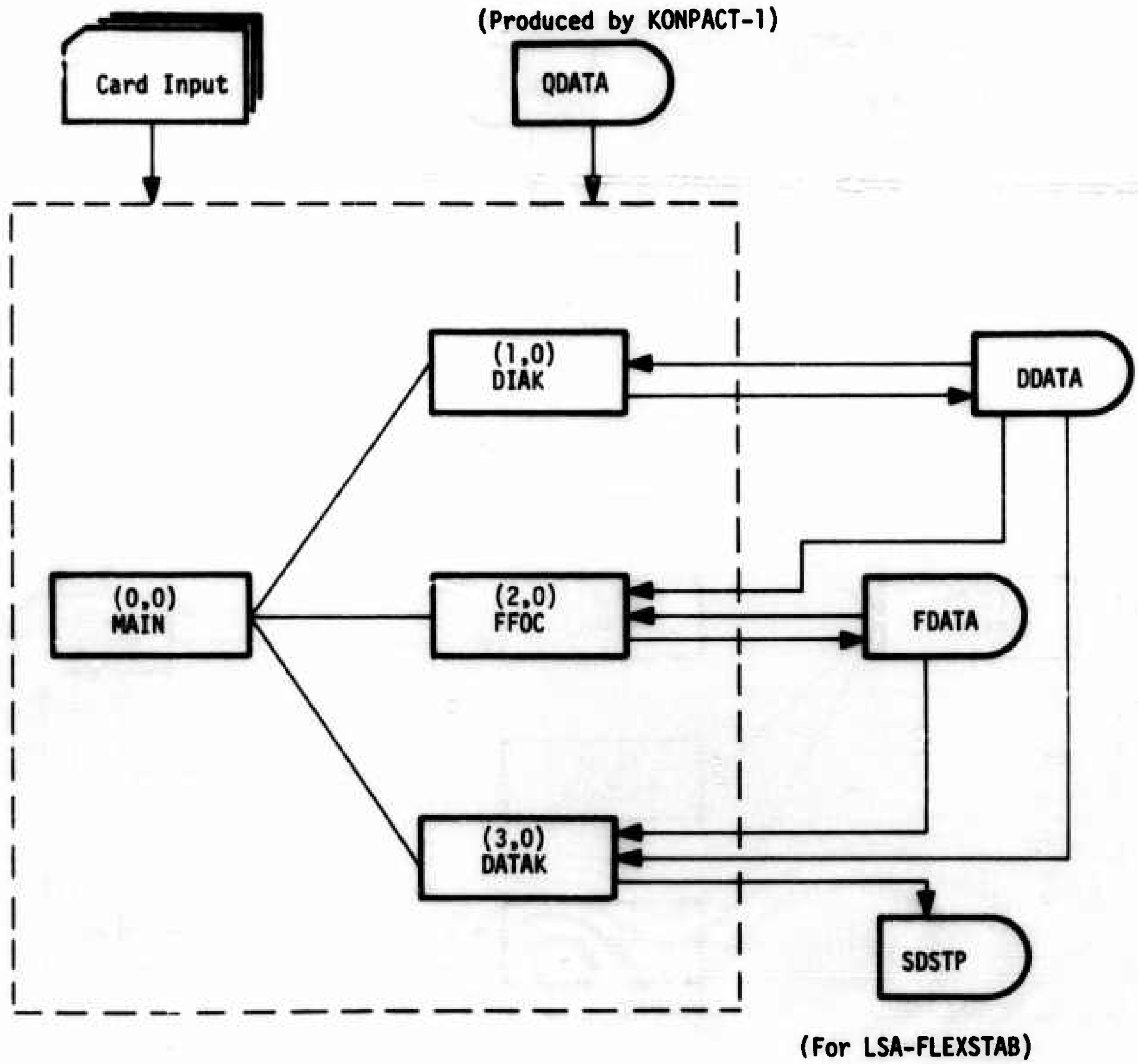

Figure 3. Overlay Structure of KONPACT-2

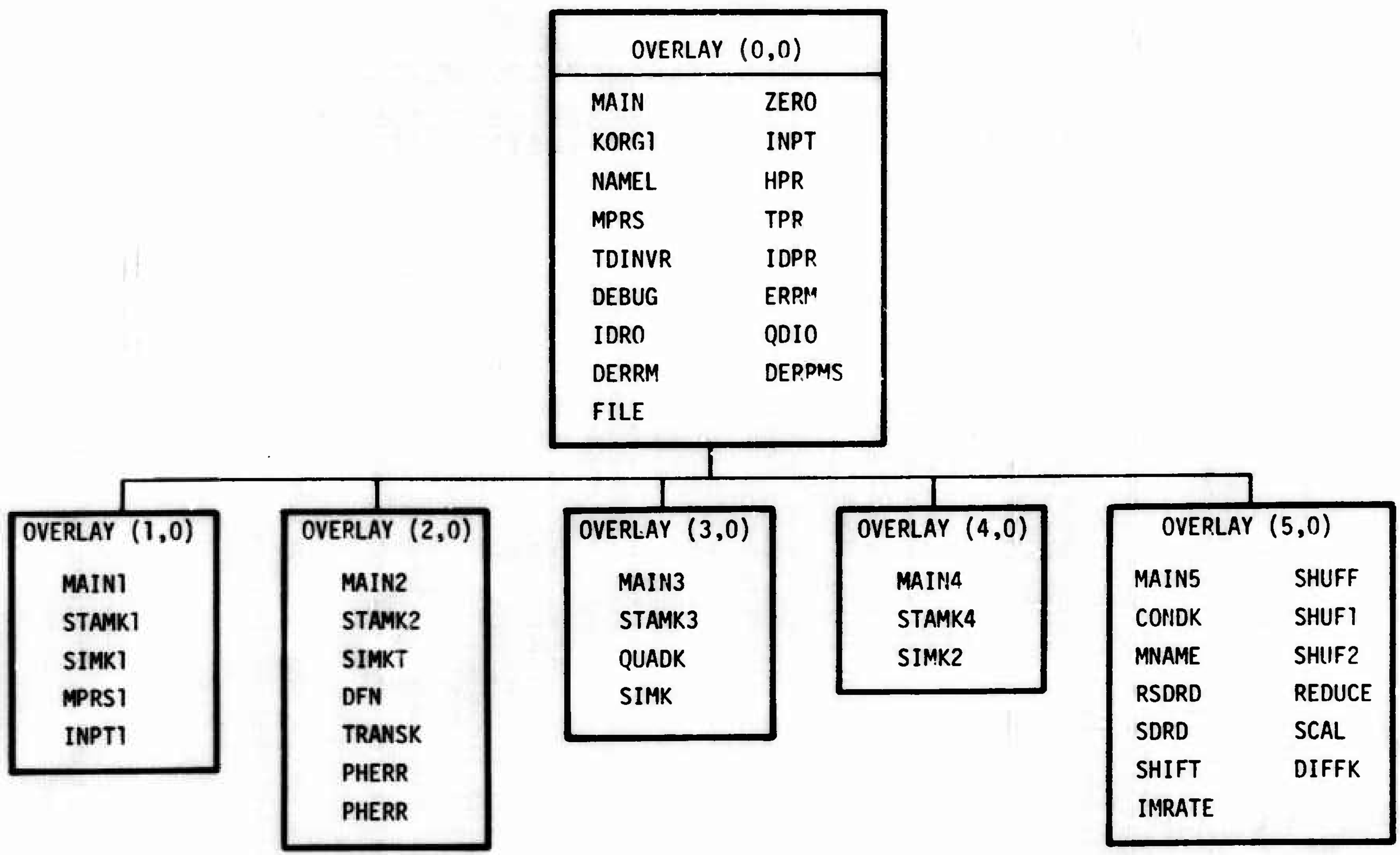

Figure 4. Overlay Structure and Subroutines in KONPACT-1

```
      OVERLAY(KOM1,0,1)
      PROGRAM MAIN(HINPUT,INPUT,NDATA,QDATA,OUTPUT,TAPE5=HINPUT,
     1TAPE6=INPUT,TAPE7=NDATA,TAPE8=QDATA,TAPE9=OUTPUT,VDATA,
     2TAPE4=VDATA,SCRTCH,TAPE3=SCRTCH)
C
C     PURPOSE - TO SET UP MAXIMUM DIMENSIONS
C     ANALISTS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE  WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        KONG1
C
C     LABELLED COMMON LIST
C        MS1                    MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C        MS2                    MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C        MS3                    MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C        MS4                    MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C        MAXN                   MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C        MAXM                   MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C        NX                     MAXIMUM NUMBER OF STATES
C        NR                     MAXIMUM NUMBER OF OUTPUTS
C        NU                     MAXIMUM NUMBER OF INPUTS
C        NY                     MAXIMUM DIMENSION FOR INTERCONN  EQUATIONS
C        MM                     MAXIMUM OF (NRM,NUM)
C        MP                     MAXIMUM DIMENSION FOR P ARRAY
C        MQ                     MAXIMUM DIMENSION FOR Q ARRAY
C        MR                     MAXIMUM DIMENSION FOR R ARRAY
C        MS (NSM)               MAXIMUM NUMBER OF SYSTEMS FOR COMBINING
C        NM                     MAXIMUM SYSTEM NUMBER - IMPLICIT MODEL
C        MS                     SAME AS NSM
C        MN                     MM*MS
C        MT (MTFB)              MAXIMUM NO OF TRANSFER FUNCTION BLOCKS
C        MST                    MAXIMUM POWER OF S IN THE TRANSFER FUNCTION
C        MT                     MAXIMUM NO OF TERMS IN THE TRANSFER FN
C        S1                     SCRATCH ARRAY FOR DYNAMIC STORAGE
C        S2                     SCRATCH ARRAY FOR DYNAMIC STORAGE
C        S3                     SCRATCH ARRAY FOR DYNAMIC STORAGE
C        S4                     SCRATCH ARRAY FOR DYNAMIC STORAGE
C
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MP,MQ,MR,MSR,NS,MS,MN,MTH,MST,MT
      COMMON /SC1/ S1(03591)
      COMMON /SC2/ S2(10000)
      COMMON /SC3/ S3(01213)
      COMMON /SC4/ S4(00001)
C
C     MAXIMUM SCRATCH ARRAY DIMENSIONS
C
      MS1=03591 $ MS2=10000 $ MS3=01213 $ MS4=00001
C
C     MAXIMUM SYSTEM DIMENSIONS
C
      NXM=1  $ NRM=1  $ NUM=10 $ NYM=30 $ MSR=3 $ MTR=10
C
C     CALL KONPACT ORGANIZING SUBROUTINE
C
      CALL KONG1
      STOP
      END
```

Figure 5. Program MAIN Program Listing

```
      OVERLAY(KONI,1   )
      PROGRAM MAIN1
C
C     PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK1
C     ANALISTS - A F KOJAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        STAMK1
C        DEERRM
C
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MP,MQ,MR,MR,NR,MS,MN,MTE8,MST,MT
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,J
      COMMON /SC1/ S1(1)
C     DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM),U(NUM)
      COMMON /SC2/ S2(1)
C     DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      COMMON /SC3/ S3(1)
C     DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
C     DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C     DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HMAIN,4H1    ,1,0,IW)
C
C     COMPUTE ARRAY START INDEXES
C
C     FOR V,W,F,U
C
      N1=1 $ N2=N1+MAXN $ N3=N2+MAXM $ N4=N3+MAXN*MAXM
      N5=N4+NUM
C
C     FOR A,B,C,D
C
      M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
      M5=M4+NRM*NUM
C
C     FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI
C
      L1=1 $ L2=L1+NXM $ L3=L2+NXM*2 $ L4=L3+NXM*10 $ L5=L4+NXM*4
      L6=L5+NRM $ L7=L6+NRM*2 $ L8=L7+NRM*10 $ L9=L8+NRM*4
      L10=L9+NUM $ L11=L10+NUM*2 $ L12=L11+NUM*10 $ L13=L12+NUM*4
C
C     CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
C
      IF((N5.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.MS3))
     1CALL DEERRM(N5,M5,L13,MS4,MS1,MS2,MS3,MS4,1,0,4HMAIN,4H1    ,IW)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HMAIN,4H1    ,1,0,IW)
C
C     CALL SUBROUTINE STAMK1
C
      CALL STAMK1(S1(N1),S1(N2),S1(N3),S1(N4),S2(M1),S2(M2),S2(M3),
     1S2(M4),S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),
     2S3(L8),S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NRM,NUM,
     3NYM,MT,MS1,MS2,MS3,MS4,NR)
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HMAIN,4H1    ,1,0,IW)
C
C     RETURN TO MAIN OVERLAY
C
      END
```

Figure 6. Program MAIN1 Program Listing

```
      OVERLAY(KON1,2,0)
      PROGRAM MAIN2
C
C     PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK2
C     ANALISTS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        STAMK2
C        DERRM
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MP,MQ,MR,MG,NR,MS,MN,MTFR,MST,MT
      COMMON /SC1/ S1(1)
C     DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
C     DIMENSION XDOT(MST,MTFR),X(MST,MTFR),RI(1,MTFR),UI(1,MTFR)
C     DIMENSION U(NUM),NNX(MTFR),NNR(MTFR),NNU(MTFR)
      COMMON /SC2/ S2(1)
C     DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
C     DIMENSION AT(MST,MST,MTFR),BT(MST,1,MTFR)
C     DIMENSION CT(1,MST,MTFR),DT(1,1,MTFR)
C     DIMENSION P(MTFR,MTFR),Q(MTFR,NUM),R(NRM,MTFR),S(NRM,NUM)
C     DIMENSION PRINT(2,MT),HS(2,MT,MTFR)
      COMMON /SC3/ S3(1)
C     DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
C     DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C     DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HMAIN,4H2    ,2,0,IW)
C
C     COMPUTE ARRAY START INDEXES
C
C     FOR V,W,F,XDOT,X,RI,UI,U,NNX,NNR,NNU
C
      N1=1 $ N2=N1+MAXN $ N3=N2+MAXM $ N4=N3+MAXN*MAXM
      N5=N4+MST*MTFR $ N6=N5+MST*MTFR $ N7=N6+MTFR $ N8=N7+MTFR
      N9=N8+NUM $ N10=N9+MTFR $ N11=N10+MTFR $ N12=N11+MTFR
C
C     FOR A,B,C,D
C
      M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
      M5=M4+NRM*NUM
C
C     FOR AT,BT,CT,DT,P,Q,R,S,PRINT,HS
C
      K1=1 $ K2=K1+MST*MST*MTFR $ K3=K2+MST*MTFR $ K4=K3+MST*MTFR
      K5=K4+MTFR $ K6=K5+MTFR*MTFR $ K7=K6+MTFR*NUM $ K8=K7+NRM*MTF
      K9=K8+NRM*NUM $ K10=K9+2*MT $ K11=K10+2*MT*MTFR
C
C     FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI
C
      L1=1 $ L2=L1+NXM $ L3=L2+NXM*2 $ L4=L3+NXM*10 $ L5=L4+NXM*4
      L6=L5+NRM $ L7=L6+NRM*2 $ L8=L7+NRM*10 $ L9=L8+NRM*4
      L10=L9+NUM $ L11=L10+NUM*2 $ L12=L11+NUM*10 $ L13=L12+NUM*4
C
C     CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
C
      MK511=M5
      IF(K11.GT.M5)MK511=K11
      IF((N12.GT.MS1).OR.(M5.GT.MS2).OR.(K11.GT.MS2).OR.(L13.GT.MS3))
     1CALL DERRM(N12,MK511,L13,MS4,MS1,MS2,MS3,MS4,2,0,4HMAIN,4H2    ,IW)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HMAIN,4H2    ,2,0,IW)
```

Figure 7. Program MAIN2 Program Listing

```
C
C     CALL SUBROUTINE STAMK2
C
      CALL STAMK2(S1(N1),S1(N2),S1(N3),S1(N4),S1(N5),S1(N6),S1(N7),
     1S1(N8),S1(N9),S1(N10),S1(N11),S2(M1),S2(M2),S2(M3),S2(M4),S2(K1),
     2S2(K2),S2(K3),S2(K4),S2(K5),S2(K6),S2(K7),S2(K8),S2(K9),S2(K10),
     3S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),S3(L8),
     4S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NRX,NUM,NYM,MH,MTFB,
     5MST,MT,MS1,MS2,MS3,MS4,NB)
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HMAIN,4H2   ,2,0,T#)
C
C     RETURN TO MAIN OVERLAY
C
      END
```

Figure 7. Program MAIN2 Program Listing (Concluded)

```
      OVERLAY(KON1,3, )
      PROGRAM MAIN3
C
C     PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK3
C     ANALISIS - A E KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEFRM
C        STAMK3
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MY,MO,MR,MB,NA,MS,MN,MTER,MST,MT
      COMMON /SC1/ S1(1)
C     DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
C     DIMENSION XDOT(NXM,MR),X(NXM,MR),RI(NRM,MR),UI(NUM,MB)
C     DIMENSION RIN(NRMMR),U(NUM),NNX(MR),NNR(MR),NNU(MB)
      COMMON /SC2/ S2(1)
C     DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      COMMON /SC3/ S3(1)
C     DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
C     DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C     DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
C
C     PRINT SYSTEM DIMENSIONS IF NEEDED
C
      IF(IPRINT.EQ.6)WRITE(IW,165)MS1,MS2,MS3,MS4,MAXN,MAXM
     1,NXM,NRM,NUM,NYM,MM,MR,MO,MR,MB,NA,MS,MN,MTER,MST,MT
  165 FORMAT(1X,15(I5,1X))
C
C     COMPUTE MAXIMUM SIZE FOR RIN
C
      NRMMR=NRM*MR
C
C     COMPUTE ARRAY START INDEXES
C
C     FOR V,W,F,XDOT,X,RI,UI,RIN,U,NNX,NNR,NNU
C
      N1=1   N2=N1+MAXN $ N3=N2+MAXM $ N4=N3+MAXN*MAXM
      N5=N4+NXM*MR $ N6=N5+NXM*MR $ N7=N6+NRM*MR $ N8=N7+NUM*MB
      N9=N8+NUM $ N10=N9+NRMMR $ N11=N10+MR $ N12=N11+MR $ N13=N12+MB
C
C     FOR A,B,C,D
C
      M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
      M5=M4+NRM*NUM
C
C     FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI
C
      L1=1 $ L2=L1+NXM $ L3=L2+NXM*2 $ L4=L3+NXM*10 $ L5=L4+NXM*4
      L6=L5+NRM $ L7=L6+NRM*2 $ L8=L7+NRM*10 $ L9=L8+NRM*4
      L10=L9+NUM $ L11=L10+NUM*2 $ L12=L11+NUM*10 $ L13=L12+NUM*4
C
C     PRINT ARRAY OVERLAPPING NUMBERS IF NEEDED
C
      IF(IPRINT.EQ.6)WRITE(IW,165)N1,N2,N3,N4,N5,N6
     1,N7,N8,N9,N10,N11,N12,N13
      IF(IPRINT.EQ.6)WRITE(IW,165)M1,M2,M3,M4,M5
      IF(IPRINT.EQ.6)WRITE(IW,165)L1,L2,L3,L4,L5,L6
     ,L7,L8,L9,L10,L11,L12,L13
C
C     CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
```

Figure 8. Program MAIN3 Program Listing

```
C                                                                       MAIN3 65
      IF((N13.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.MS3))                   MAIN3 66
     1CALL ERRM(N13,MK5S,L13,MS4,MS1,MS2,MS3,MS4,3,1,4HMAIN,4H3    ,IW) MAIN3 67
C                                                                       MAIN3 68
C     CALL SUBROUTINE STAMK3                                            MAIN3 69
C                                                                       MAIN3 70
      CALL STAMK3(S1(N1),S1(N2),S1(N3),S1(N4),S1(N5),S1(N6),S1(N7),     MAIN3 71
     1S1(N8),S1(N9),S1(N10),S1(N11),S1(N12),S2(M1),S2(M2),S2(M3),S2(M4),MAIN3 72
     2S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),S3(L8),          MAIN3 73
     3S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NZM,NUM,NYM,MH,      MAIN3 74
     4MN,MM,MP,MQ,MR,MS1,MS2,MS3,M5,NH,NRMMH)                           MAIN3 75
C                                                                       MAIN3 76
C     RETURN TO MAIN OVERLAY                                            MAIN3 77
C                                                                       MAIN3 78
      END                                                               MAIN3 79
```

Figure 8. Program MAIN3 Program Listing (Concluded)

```
      OVERLAY(KON1,4,0)
      PROGRAM MAIN4
C
C     PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK4
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DERRM
C        STAMK4
C
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MP,MQ,MR,MB,NB,MS,MN,MTFB,MST,MT
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
      COMMON /SC1/ S1(1)
C     DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM),U(NUM)
      COMMON /SC2/ S2(1)
C     DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      COMMON /SC3/ S3(1)
C     DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
C     DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C     DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
C
C     COMPUTE ARRAY START INDEXES
C
C     FOR V,W,F,U
C
      N1=1 $ N2=N1+MAXN $ N3=N2+MAXM $ N4=N3+MAXN*MAXM
      N5=N4+NUM
C
C     FOR A,B,C,D
C
      M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
      M5=M4+NRM*NUM
C
C     FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI
C
      L1=1 $ L2=L1+NXM $ L3=L2+NXM*2 $ L4=L3+NXM*10 $ L5=L4+NXM*4
      L6=L5+NRM $ L7=L6+NRM*2 $ L8=L7+NRM*10 $ L9=L8+NRM*4
      L10=L9+NUM $ L11=L10+NUM*2 $ L12=L11+NUM*10 $ L13=L12+NUM*4
C
C     CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
C
      IF((N5.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.MS3))
     1CALL DERRM(N5,M5,L13,MS4,MS1,MS2,MS3,MS4,4,0,4HMAIN,4H4    ,IW)
C
C     CALL SUBROUTINE STAMK4
C
      CALL STAMK4(S1(N1),S1(N2),S1(N3),S1(N4),S2(M1),S2(M2),S2(M3),
     1S2(M4),S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),
     2S3(L8),S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NRM,NUM,
     3NYM,M4,MS1,MS2,MS3,MS4,NB)
C
C     RETURN TO MAIN OVERLAY
C
      END
```

Figure 9. Program MAIN4 Program Listing

```
      OVERLAY(KONI,5,0)
      PROGRAM MAIN5

      PURPOSE - TO SET UP DIMENSIONS AND CALL CONOK
      ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
      DATE WRITTEN - 1975

      SUBPROGRAMS CALLED
          DERRM
          RESPK
          DEBUG

      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MP,MQ,MR,MB,NH,MS,MN,MTEB,MST,MT
      COMMON /SYS/ SCODE,SDES(5),NSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      COMMON /SC2/ S2(1)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION CM(NRM,NYM),DM(NRM,NUM)
      COMMON /SC3/ S3(1)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DIMENSION NNNS(NXM),VNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4)
      DIMENSION NNNO(NRM),VNNO(NRM,2),DESNO(NRM,10),UNITNO(NRM,4)
      DIMENSION NNNI(NUM),VNNI(NUM,2),DESNI(NUM,10),UNITNI(NUM,4)
      COMMON /SC1/ S1(1)
      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22),DUMMY3(NUM)
      DIMENSION ES(NXM,NUM),ER(NRM,NUM)
      DIMENSION NSHUFS(NXM),NSHUFO(NRM),NSHUFI(NUM)
      DIMENSION CS(NRM,NXM),DS(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
      DIMENSION IRS(NRM),Q(NRM,NRM)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HMAIN,4H5   ,5,0,IW)
      NXUM=NXM+NUM $ MWORD=17 $ NRSM=1
      NDM11=MAX0(MWORD,NXM,NRM,NRSM)
      NDM12=MAX0(NXUM,NRM)
      NDM21=MAX0(NRM,NXM,NRSM)
      NDM22=MAX0(NXM,NUM,NRM)
C
C     PRINT ERROR MESSEGE IF DIMENSION OF SCRATCH ARRAYS ARE INSUFFICIEN
C
      M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
      M5=M4+NRM*NUM $ M6=M5+NRM*NXM $ M7=M6+NRM*NUM
      N1=1 $ N2=N1+NXM $ N3=N2+NXM*2 $ N4=N3+NXM*10 $ N5=N4+NXM*4
      N6=N5+NRM $ N7=N6+NRM*2 $ N8=N7+NRM*10 $ N9=N8+NRM*4
      N10=N9+NUM $ N11=N10+NUM*2 $ N12=N11+NUM*10 $ N13=N12+NUM*4
      N14=N13+NXM $ N15=N14+NXM*2 $ N16=N15+NXM*10 $ N17=N16+NXM*4
      N18=N17+NRM $ N19=N18+NRM*2 $ N20=N19+NRM*10 $ N21=N20+NRM*4
      N22=N21+NUM $ N23=N22+NUM*2 $ N24=N23+NUM*10 $ N25=N24+NUM*4
      L1=1 $ L2=L1+NDM11*NDM12 $ L3=L2+NDM21*NDM22 $ L4=L3+NUM
      L5=L4+NXM*NUM $ L6=L5+NRM*NUM $ L7=L6+NXM $ L8=L7+NRM
      L9=L8+NUM $ L10=L9+NRM*NXM $ L11=L10+NRM*NUM
      L12=L11+NRM*NXM $ L13=L12+NRM*NUM $ L14=L13+NRM
      L15=L14+NRM*NRM
      IF((L15.GT.MS1).OR.(M7.GT.MS2).OR.(N25.GT.MS3))
     1CALL DERRM(L15,M7,N25,MS4,MS1,MS2,MS3,MS4,5,0,4HMAIN,4H5   ,IW)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HMAIN,4H5   ,5,0,IW)
C
C     CALL SUBROUTINE CONOK
C
      CALL CONOK(S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),
     1S3(N1),S3(N2),S3(N3),S3(N4),S3(N5),S3(N6),
```

Figure 10. Program MAIN5 Program Listing

```
     2S3(N7),S3(N8),S3(N9),S3(N10),S3(N11),S3(N12),
     3S3(N13),S3(N14),S3(N15),S3(N16),S3(N17),S3(N18),
     4S3(N19),S3(N20),S3(N21),S3(N22),S3(N23),S3(N24),
     5S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
     6S1(L7),S1(L8),S1(L9),S1(L10),S1(L11),S1(L12),
     7S1(L13),S1(L14),NXM,NRM,NUM,NDM11,NDM12,NDM21,NDM22)
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HMAIN,4H5   ,5,0,IW)
C
C     RETURN TO MAIN OVEPLAY
C
      END
```

Figure 10. Program MAIN5 Program Listing (Concluded)

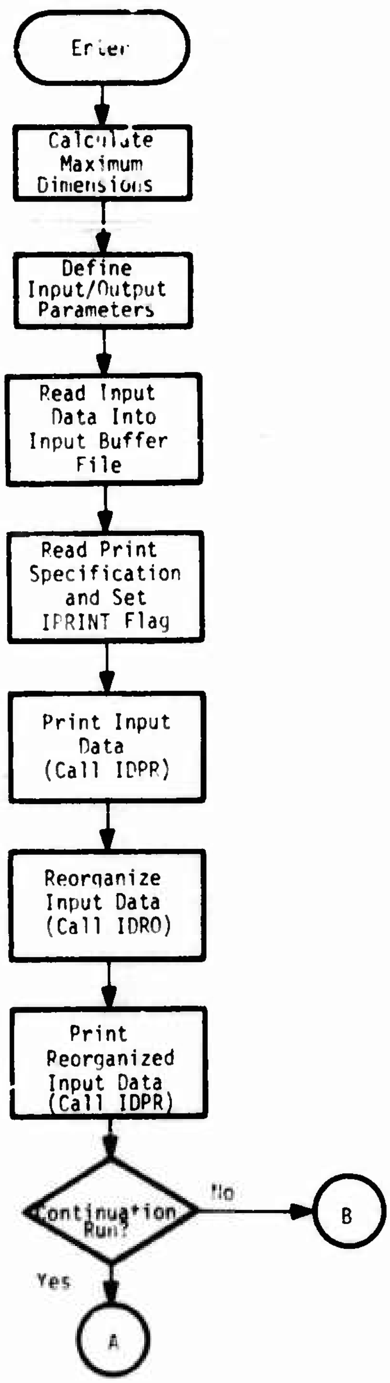

Figure 11. Subroutine KORG1 Flow Chart

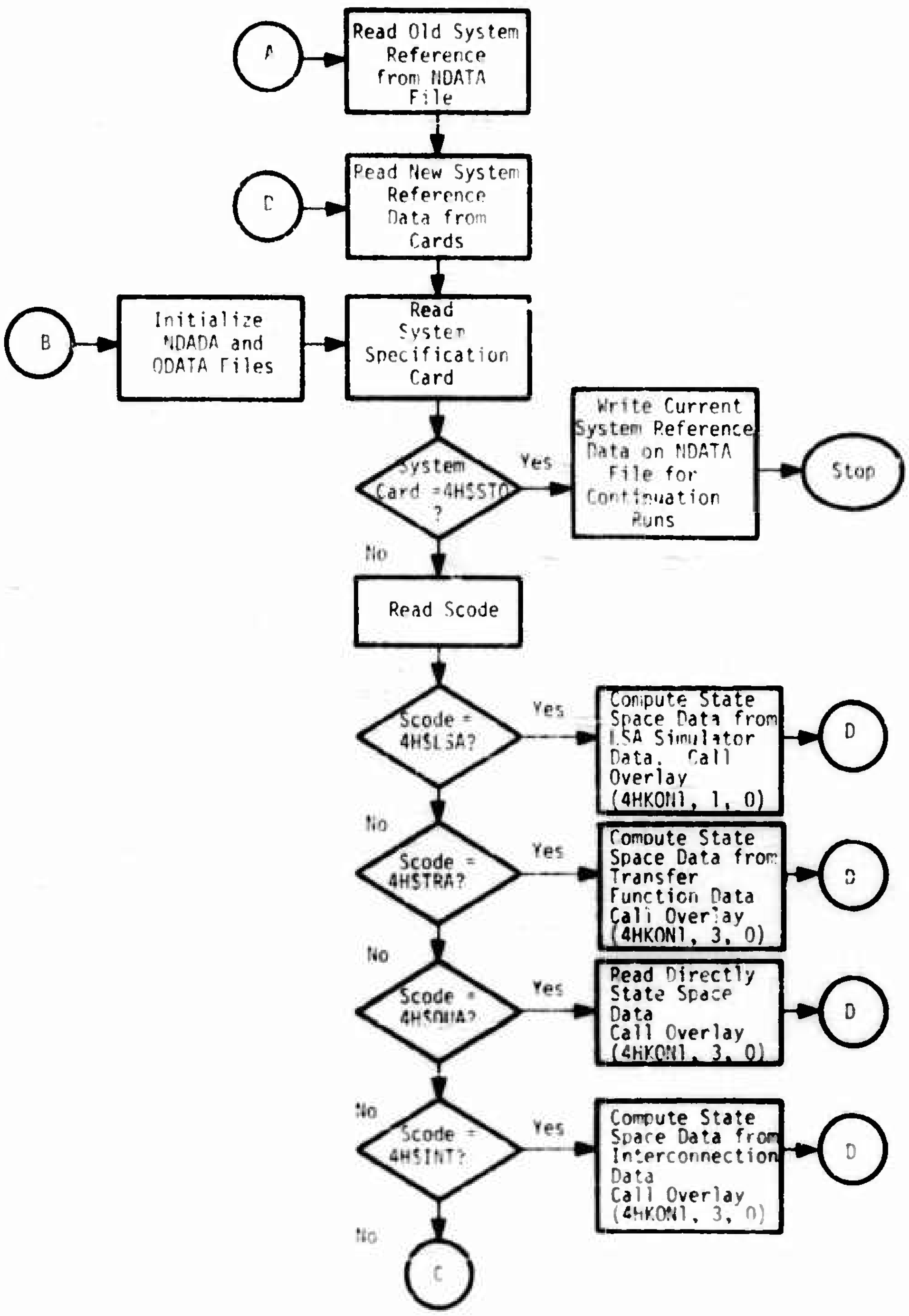

Figure 11. Subroutine KORG1 Flow Chart (Continued)

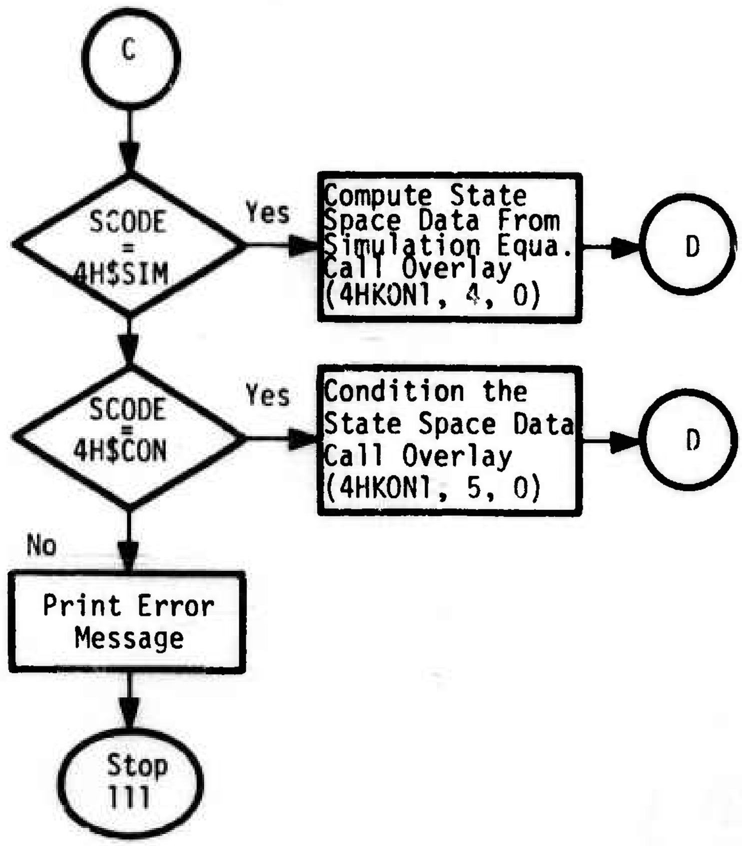

Figure 11. Subroutine KORG1 Flow Chart (Concluded)

```
      SUBROUTINE KORG1
C
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     PURPOSE - TO ORGANIZE EXECUTION OF KONPACT-1 PROGRAMS
C     DATE WRITTEN - JULY 1975
C
C     SUBPROGRAMS CALLED
C        HPR
C        IDRO
C        IDRP
C        FILE
C
C     LABELLED COMMON LIST
C        IR                    FILE NUMBER FOR CARD READER
C        IW                    FILE NUMBER FOR LINE PRINTER
C        IPRINT                PRINT CONTROL FLAG
C        INSERT                HOLLERITH INSE
C        LOCATE                HOLLERITH LOCA
C        NULL                  HOLLERITH NULL
C        MARK                  HOLLERITH $$..$
C        JN                    FILE NO FOR NAME LIST DATA FILE
C        JQ                    FILE NO FOR QUADRUPLE DATA FILE
C        JS                    FILE NO FOR SCRATCH FILE
C        IHEAD                 LABEL NAME
C        SCODE                 SYSTEM CODE WORD
C        SDES                  SYSTEM DESCRIPTION
C        MSYS                  PRESENT SYSTEM NUMBER
C        HEAD                  PRESENT SYSTEM HEADING
C        NSYS                  SYSTEM NUMBERS
C        SHEAD                 SYSTEM HEADINGS
C        PHEAD                 PREVIOUS SYSTEM HEADING
C
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
     1,MM,MP,MQ,MR,MR,NB,MS,MN,MTFB,MST,MT
      DIMENSION CARD(20),LABEL(20),AHEAD(20,20)
      INTEGER HINSE,HLOCA,HNULL,HDOLR
      DATA HINSE,HLOCA,HNULL,HDOLR/4HINSE,4HLOCA,4HNULL,4H$$$$/
      DATA HSTOP,HSYST,HEM,HC,HCONT/4HSTOP,4HSYST,4HEM  ,2HC ,4HCONT/
      DATA HPRIN,HTHIN,HERYT/4HPRIN,4HTHIN,4HERYT/
      DATA HTPUT,HNAL,HPUT/4HTPUT,4HNAL ,4HPUT /
      DATA HDLSA,HDCON,HDQUA/4H$LSA,4H$CON,4H$QUA/
      DATA HDTRA,HDINT,HDSIM/4H$TRA,4H$INT,4H$SIM/
      DATA HREFE,HRENC,HEBBB,HBBBB/4HREFE,4HRENC,4HE   ,4H    /
      DATA HEND/4HEND /
C
C     MAXIMUM DIMENSIONS FOR V,W AND F
C
      MAXN=NXM+NYM+NRM $ MAXM=NXM+NYM+NXM+NJM
C
C     MAXIMUM DIMENSIONS FOR COMBINING TRANSFER FUNCTION BLOCKS
C
      MST=5 $ MT=6
C
C     MAXIMUM DIMENSIONS FOR COMBINING SUBSYSTEMS
C
      MM=NRM
      IF(MM.LT.NUM)MM=NUM
      MS=NXM $ MQ=MR $ MR=MR $ MP=MM*2 $ MN=MM*MR
C
C     MAXIMUM SYSTEM NUMBER
```

Figure 12. Subroutine KORG1 Program Listing

```
C
      NB=9
C
C     DEFINE INPUT/OUTPUT PARAMETERS
C
      IR=5 $ IW=9 $ IPRINT=4 $ JN=7 $ JQ=8 $ JS=3
     ' INSERT=HINSE $ LOCATE=HLOCA $ NULL=HNULL
      DO 10  I=1,20
  100 MARK(I)=HDOLR
      LABEL(1)=HREFE
      LABEL(2)=HRENC
      LABEL(3)=HEBBB
      DO 105 I=4,20
  105 LABEL(I)=HBBBB
C
C     READ INPUT DATA INTO INPUT DATA BUFFER FILE
C
      REWIND IR
  110 CONTINUE
      READ(6,120)CARD
      IF(EOF(6))140,115
  115 CONTINUE
      WRITE(IR,120)CARD
  120 FORMAT(20A4)
      GO TO 110
  140 CONTINUE
      ENDFILE IR
      REWIND IR
C
C     READ PRINT SPECIFICATION AND SET IPRINT
C
  142 CONTINUE
      READ(IR,170)CARD
      DECODE(4,143,CARD(1))CC,DUMMY
  143 FORMAT(A2,A2)
      IF(CC.EQ.HC)GO TO 142
      IF(CARD(1).NE.HPRIN)GO TO 152
      IF(CARD(3).EQ.HTHIN)IPRINT=5
      IF(CARD(3).EQ.HTHIN)GO TO 142
      IF(CARD(3).EQ.HERYT)IPRINT=6
      IF(CARD(3).EQ.HERYT)GO TO 142
      IF(CARD(3).NE.HTPUT)GO TO 144
      IF(IPRINT.EQ.1)IPRINT=5
      IF(IPRINT.EQ.5)GO TO 142
      IPRINT=3
      GO TO 142
  144 CONTINUE
      IF(CARD(3).NE.HNAL)GO TO 146
      IF(IPRINT.EQ.1)IPRINT=4
      IF(IPRINT.EQ.4)GO TO 142
      IPRINT=2
      GO TO 142
  146 CONTINUE
      IF(CARD(3).NE.HPUT)GO TO 148
      IF(IPRINT.EQ.6)IPRINT=1
      IF(IPRINT.EQ.1)GO TO 142
      IF(IPRINT.EQ.2)IPRINT=4
      IF(IPRINT.EQ.3)IPRINT=5
      IF(IPRINT.EQ.4)GO TO 142
      IF(IPRINT.EQ.5)GO TO 142
      IPRINT=1
      GO TO 142
C
C     PRINT ERROR MESSEGE
C
  148 CONTINUE
```

Figure 12. Subroutine KORG1 Program Listing (Continued)

```
      WRITE(IW,150)
  150 FORMAT(1H1,//,1X,30HPRINT CARD SPECIFICATION ERROR,//,1X,
     143HINPUT AND FINAL OUTPUT DATA WILL BE PRINTED)
      IPRINT=4
C
C     PRINT INPUT DATA
C
  152 CONTINUE
      REWIND IR
      IF((IPRINT.NE.1).AND.(IPRINT.LT.4))GO TO 158
      WRITE(IW,154)
  154 FORMAT(1H1,//,1X,24H*** INPUT DATA CARDS ***,//)
      CALL IDPR(IR,IW)
      REWIND IR
  158 CONTINUE
C
C     REORGANIZE INPUT DATA
C
      CALL IDRO(IR,IW,JS)
C
C     PRINT REORGANIZED INPUT DATA
C
      IF(IPRINT.LT.6)GO TO 164
      WRITE(IW,160)
  160 FORMAT(1H1,//,1X,30H*** REORGANIZED INPUT DATA ***,//)
      CALL IDPR(IR,IW)
C
C     READ INITIALIZING INSTRUCTIONS
C
  164 CONTINUE
      ISYS=0
      DO 166 I=1,9
      DO 166 J=1,20
  166 SHEAD(I,J)=HBBBB
  168 CONTINUE
      READ(IR,170)CARD
  170 FORMAT(20A4)
      IF(CARD(1).EQ.HPRIN)GO TO 168
      IF(CARD(1).NE.HCONT)GO TO 175
      CALL FILE(JN,LOCATE,LABEL)
      READ(JN)((SHEAD(I,J),J=1,20),I=1,9)
      CALL FILE(JN,NULL,LABEL)
      WRITE(IW,430)
      WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)
      GO TO 180
  175 CONTINUE
      CALL FILE(JN,INSERT,MARK)
      CALL FILE(JO,INSERT,MARK)
      GO TO 190
C
C     READ SYSTEM REFERENCE DATA
C
  180 CONTINUE
      READ(IR,170)CARD
      IF(CARD(1).NE.HREFE)GO TO 190
  183 CONTINUE
      READ(IR,170)CARD
      IF(CARD(1).EQ.HEND)GO TO 188
      DECODE(4,220,CARD(3))D1,NSYSNO,D2
      DO 185 I=1,20
  185 SHEAD(NSYSNO,I)=CARD(I)
      GO TO 183
  188 CONTINUE
      WRITE(IW,430)
      WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)
C
```

Figure 12. Subroutine KORG1 Program Listing (Continued)

```
C     READ SYSTEM SPECIFICATION CARD
C
      READ(IR,170)CARD
  190 CONTINUE
      IF(CARD(1).EQ.HSTOP)GO TO 400
      IF(CARD(1).NE.HSYST)GO TO 260
      IF(IPRINT.LT.6)GO TO 210
      CALL HPR(CARD,IW)
      WRITE(IW,200)MS1,MS2,MS3,MS4,MAXN,MAXM
     1,NXM,NRM,NUM,NYM,MM,MP,MQ,MR,MB,NB,MS,MN,MTFB,MST,MT
  200 FORMAT(1X,15(I5,1X))
  210 CONTINUE
      DECODE(4,220,CARD(3))D1,NSYSNO,D2
  220 FORMAT(A2,I1,A1)
      IF(NSYSNO.GT.NB)GO TO 260
      ISYS=ISYS+1
      IF(ISYS.GT.20)GO TO 260
      DO 240 I=1,5
      II=5+I
  240 SDES(I)=CARD(II)
      DO 245 I=1,20
  245 PHEAD(I)=SHEAD(NSYSNO,I)
      DO 250 I=1,20
      HEAD(I)=CARD(I)
      AHEAD(ISYS,I)=CARD(I)
  250 SHEAD(NSYSNO,I)=CARD(I)
      NSYS(ISYS)=NSYSNO
      IF(IPRINT.LT.6)GO TO 256
      WRITE(IW,253)CARD
      WRITE(IW,253)HEAD
      WRITE(IW,253)PHEAD
      WRITE(IW,253)(SHEAD(NSYSNO,I),I=1,20)
  253 FORMAT(1X,20A4)
  256 CONTINUE
      READ(IR,170)SCODE
      IF(SCODE.EQ.HDLSA)GO TO 300
      IF(SCODE.EQ.HDTRA)GO TO 320
      IF(SCODE.EQ.HDQUA)GO TO 340
      IF(SCODE.EQ.HDINT)GO TO 340
      IF(SCODE.EQ.HDSIM)GO TO 360
      IF(SCODE.EQ.HDCON)GO TO 380
C
C     PRINT ERROR MESSEGE
C
  260 CONTINUE
      WRITE(IW,280)
  280 FORMAT(1H1,//,1X,31HSYSTEM CARD SPECIFICATION ERROR)
      WRITE(IW,290)CARD
      WRITE(IW,290)SCODE
  290 FORMAT(1X,20A4)
      WRITE(IW,295)NSYSNO,NB
  295 FORMAT(1X,I2,1X,I2)
      STOP 111
C
C     CALL OVERLAY LOADER TO LOAD REQUIRED PROGRAMS FOR EXECUTION
C
  300 CONTINUE
      CALL OVERLAY(4HKON1,1,0)
      GO TO 180
  320 CONTINUE
      CALL OVERLAY(4HKON1,2,0)
      GO TO 180
  340 CONTINUE
      CALL OVERLAY(4HKON1,3,0)
      GO TO 180
  360 CONTINUE
```

Figure 12. Subroutine KORG1 Program Listing (Continued)

```
      CALL OVERLAY(4HKON1,4,0)                                          KORG1263
      GO TO 180                                                         KORG1264
  380 CONTINUE                                                          KORG1265
      CALL OVERLAY(4HKON1,5,0)                                          KORG1266
      GO TO 180                                                         KORG1267
C                                                                       KORG1268
C     WRITE SYSTEM LABELS ON NFILE FOR CONTINUATION RUNS                KORG1269
C                                                                       KORG1270
  400 CONTINUE                                                          KORG1271
      CALL FILE(JN,INSERT,LABEL)                                        KORG1272
      WRITE(JN)((SHEAD(I,J),J=1,20),I=1,9)                              KORG1273
      CALL FILE(JN,INSERT,MARK)                                         KORG1274
      WRITE(IW,430)                                                     KORG1275
  430 FORMAT(1H1,//,1X,34H*** REFERENCE OF SYSTEM LABELS ***,//)        KORG1276
      WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)                          KORG1277
  440 FORMAT(/,1X,20A4,/)                                               KORG1278
      WRITE(IW,450)                                                     KORG1279
  450 FORMAT(1H1,//,1X,41H*** LIST OF SYSTEM LABELS CREATED IN THIS,    KORG1280
     18H RUN ***,//)                                                    KORG1281
      WRITE(IW,440)((AHEAD(I,J),J=1,20),I=1,ISYS)                       KORG1282
      STOP                                                              KORG1283
      END                                                               KORG1284
```

Figure 12. Subroutine KORG1 Program Listing (Concluded)

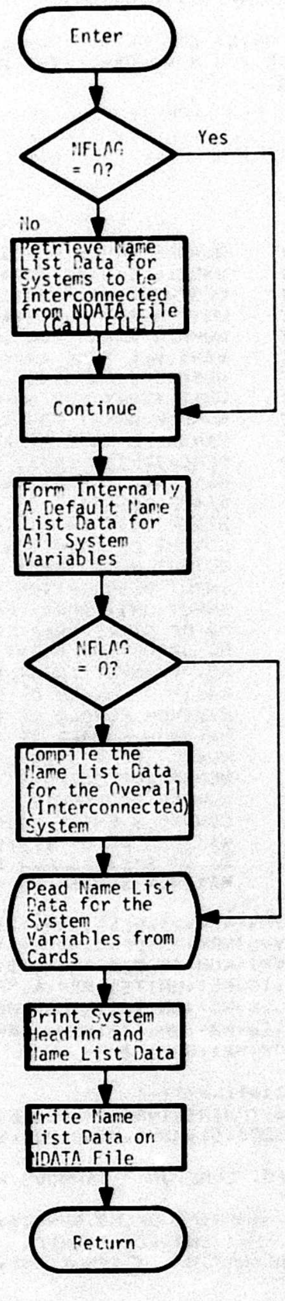

Figure 13. Subroutine NAMEL Flow Chart

```
      SUBROUTINE NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
     1DESI,UNITI,DESSS,UNITSS,DESOO,UNITOO,DESII,UNITII,NXX,NRR,NUU,
     2NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)
C
C     PURPOSE - TO READ, PRINT AND UPDATE NAMELIST DATA FOR SYSTEMS
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        HPR
C        FILE
C
C     ARGUMENTS LIST
C        NNS        IN/OUT    NUMBER ARRAY FOR STATE
C        VNS        IN/OUT    VARIABLE NAME ARRAY FOR STATE
C        DESS       IN/OUT    DESCRIPTION ARRAY FOR STATE
C        UNITS      IN/OUT    UNIT ARRAY FOR STATE
C        NNO        IN/OUT    NUMBER ARRAY FOR OUTPUT
C        VNO        IN/OUT    VARIABLE NAME ARRAY FOR OUTPUT
C        DESO       IN/OUT    DESCRIPTION ARRAY FOR OUTPUT
C        UNITO      IN/OUT    UNIT ARRAY FOR OUTPUT
C        NNI        IN/OUT    NUMBER ARRAY FOR INPUT
C        VNI        IN/OUT    VARIABLE NAME ARRAY FOR INPUT
C        DESI       IN/OUT    DESCRIPTION ARRAY FOR INPUT
C        UNITI      IN/OUT    UNIT ARRAY FOR INPUT
C        DESSS                STATE DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C        UNITSS               STATE UNIT ARRAY FOR ALL SUBSYSTEMS
C        DESOO                OUTPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C        UNITOO               OUTPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C        DESII                INPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C        UNITII               INPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C        NXX                  NO OF STATE ARRAY FOR ALL SUBSYSTEMS
C        NRR                  NO OF OUTPUT ARRAY FOR ALL SUBSYSTEMS
C        NUU                  NO OF INPUT ARRAY FOR ALL SUBSYSTEMS
C        NXM        INPUT     MAXIMUM NUMBER OF STATES
C        NRM        INPUT     MAXIMUM NUMBER OF OUTPUTS
C        NUM        INPUT     MAXIMUM NUMBER OF INPUTS
C        NX         INPUT     NUMBER OF STATES
C        NR         INPUT     NUMBER OF OUTPUTS
C        NU         INPUT     NUMBER OF INPUTS
C        NFLAG      INPUT     CONTROLS ENTRY POINT IN THE SUBROUTINE
C        MB         INPUT     MAXIMUM NO OF SYSTEMS FOR COMBINING
C        KB         OUTPUT    NO OF SYSTEMS FOR COMBINING
C        NB         INPUT     MAXIMUM SYSTEM NO - IMPLICIT MODEL
C
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB)
      DIMENSION DESOO(NRM,10,MB),UNITOO(NRM,4,MB)
      DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)
      DIMENSION NXX(MB),NRR(MB),NUU(MB)
      DIMENSION CARD(20)
      DIMENSION VN(2),DES(10),UNIT(4)
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      DATA HBB9,HMODE,HLBFO,HLLBE/3H   ,4HMODE,4HL FO,4HLL E/
      DATA HRROR/4HRROR/
      DATA HC,HSTAT,HOUTP,HINPU,HEND/1HC,4HSTAT,4HOUTP,4HINPU,4HEND /
      DATA HXP,HRP,HUP,HP/2HX(,2HR(,2HU(,1H)/
      DATA HB,HE,HUT,HT,HBLANK/1H ,1HE,2HUT,1HT,4H    /
```

Figure 14. Subroutine NAMEL Program Listing

```
      REWIND JN
      NRI=0
      IF(NFLAG.EQ.0)GO TO 380
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HNAME,4HL    ,0,0,IW)
C
C     RETRIEVE NAME LIST DATA OF SUBSYSTEMS FOR COMBINING FROM FILE
C     NDATA
C
      DO 10 I=1,20
   10 CARD(I)=HEAD(I)
      DO 120 N=1,KB
      NSY=NSYS(N)
      DO 40 I=1,20
   40 HEAD(I)=SHEAD(NSY,I)
      CALL FILE(JN,LOCATE,HEAD)
      READ(JN)NXN,NRN,NUN,
     1          (NNS(I),(VNS(I,J),J=1,2),
     2          (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NXN),
     3          (NNO(I),(VNO(I,J),J=1,2),
     4          (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NRN),
     5          (NNI(I),(VNI(I,J),J=1,2),
     6          (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NUN)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HNAME,4HL    ,0,0,IW)
      NXX(N)=NXN
      NRR(N)=NRN
      NUU(N)=NUN
C
C     IF THE SUBSYSTEM IS AN IMPLICIT MODEL, THEN SET NRI=NRN
C
      IF(NSY.EQ.NB)NRI=NRN
      DO 60 I=1,NXN
      DO 50 J=1,10
   50 DESSS(I,J,N)=DESS(I,J)
      DO 60 J=1,4
   60 UNITSS(I,J,N)=UNITS(I,J)
      DO 80 I=1,NRN
      DO 70 J=1,10
   70 DESOO(I,J,N)=DESO(I,J)
      DO 80 J=1,4
   80 UNITOO(I,J,N)=UNITO(I,J)
      DO 100 I=1,NUN
      DO 90 J=1,10
   90 DESII(I,J,N)=DESI(I,J)
      DO 100 J=1,4
  100 UNITII(I,J,N)=UNITI(I,J)
  120 CONTINUE
      DO 130 I=1,20
  130 HEAD(I)=CARD(I)
  380 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HNAME,4HL    ,0,0,IW)
C
C     FORM A DEFAULT NAME LIST TABLE FOR THE SYSTEM
C
C     FORM NAME LIST FOR STATES
C
      DO 500 II=1,NX
      NNS(II)=II
      ENCODE(4,420,VNS(II,1))HXP,II
  420 FORMAT(A2,I2)
      VNS(II,2)=HP
      UNITS(II,1)=HBLANK
      UNITS(II,2)=HBLANK
      UNITS(II,3)=HBLANK
      UNITS(II,4)=HBLANK
      J=0
      JFLAG=0
```

Figure 14. Subroutine NAMEL Program Listing (Continued)

```
 440 CONTINUE
     IF(J.GE.5)GO TO 460
     IF(JFLAG.EQ.1)GO TO 460
     J=J+1
     IF(SDES(J).EQ.HBLANK)JFLAG=JFLAG+1
     DESS(II,J)=SDES(J)
     GO TO 440
 460 CONTINUE
     J=J+1
     DESS(II,J)=HSTAT
     J=J+1
     ENCODE(4,470,DESS(II,J))HE,II,HB
 470 FORMAT(A1,I2,A1)
 480 CONTINUE
     J=J+1
     IF(J.GT.10)GO TO 500
     DESS(II,J)=HBLANK
     GO TO 480
 500 CONTINUE
     IF(IPRINT.EQ.6)CALL DEBUG(4,4HNAME,4HL    ,0,6,TW)
C
C    FORM NAME LIST FOR OUTPUTS
C
     NRJ=NR-NRI
     DO 700 II=1,NR
     NNO(II)=II
     ENCODE(4,420,VNO(II,1))HRP,II
     VNO(II,2)=HP
     UNITO(II,1)=HBLANK
     UNITO(II,2)=HBLANK
     UNITO(II,3)=HBLANK
     UNITO(II,4)=HBLANK
     J=0
     JFLAG=0
C
C    FORM NAME LIST FOR THE IMPLICIT MODEL ERROR RESPONSES
C
     IF(II.LE.NRJ)GO TO 640
     NJ=II-NRJ
     DECODE(4,610,DESOO(NJ,1,KB))IT1,JT1
 610 FORMAT(A3,A1)
     ENCODE(4,615,DESO(II,1))HE,IT1
 615 FORMAT(A1,A3)
     ENCODE(4,615,DESO(II,2))JT1,HBBB
     DESO(II,3)=HMODE
     DESO(II,4)=HLBEO
     DESO(II,5)=HLLBE
     DESO(II,6)=HRROR
     DESO(II,7)=HBLANK
     DESO(II,8)=HBLANK
     DESO(II,9)=HBLANK
     DESO(II,10)=HBLANK
     DO 620 J=1,4
 620 UNITO(II,J)=UNITOO(NJ,J,KB)
     GO TO 700
 640 CONTINUE
     IF(J.GE.5)GO TO 660
     IF(JFLAG.EQ.1)GO TO 660
     J=J+1
     IF(SDES(J).EQ.HBLANK)JFLAG=JFLAG+1
     DESO(II,J)=SDES(J)
     GO TO 640
 660 CONTINUE
     J=J+1
     DESO(II,J)=HOUTP
     J=J+1
```

Figure 14. Subroutine NAMEL Program Listing (Continued)

```
      ENCODE(4,420,DESO(II,J))HUT,II
  680 CONTINUE
      J=J+1
      IF(J.GT.10)GO TO 700
      DESO(II,J)=HBLANK
      GO TO 680
  700 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HNAME,4HL    ,0,0,IW)
C
C     FORM NAME LIST FOR INPUTS
C
      DO 900 II=1,NU
      NNI(II)=II
      ENCODE(4,420,VNI(II,1))HUP,II
      VNI(II,2)=HP
      UNITI(II,1)=HBLANK
      UNITI(II,2)=HBLANK
      UNITI(II,3)=HBLANK
      UNITI(II,4)=HBLANK
      J=0
      JFLAG=0
  840 CONTINUE
      IF(J.GE.5)GO TO 860
      IF(JFLAG.EQ.1)GO TO 860
      J=J+1
      IF(SDES(J).EQ.HBLANK)JFLAG=JFLAG+1
      DESI(II,J)=SDES(J)
      GO TO 840
  860 CONTINUE
      J=J+1
      DESI(II,J)=HINPU
      J=J+1
      ENCODE(4,470,DESI(II,J))HT,II,HB
  880 CONTINUE
      J=J+1
      IF(J.GT.10)GO TO 900
      DESI(II,J)=HBLANK
      GO TO 880
  900 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(6,4HNAME,4HL    ,0,0,IW)
      IF(NFLAG.EQ.0)GO TO 1220
C
C     COMBINE THE NAME LIST DATA OF SUBSYSTEMS AND OBTAIN THE NAME LIST
C     DATA FOR THE COMBINED SYSTEM
C
 1000 CONTINUE
      II=0
      DO 1040 K=1,KB
      NXXK=NXX(K)
      DO 1040 I=1,NXXK
      II=II+1
      NNS(II)=II
      ENCODE(4,420,VNS(II,1))HXP,II
      VNS(II,2)=HP
      DO 1020 J=1,10
 1020 DESS(II,J)=DESSS(I,J,K)
      DO 1040 J=1,4
 1040 UNITS(II,J)=UNITSS(I,J,K)
C
C     READ NAME LIST DATA FOR OUTPUTS OBTAINABLE FROM
C     INTERCONNECTION EQUATIONS WRITTEN ON SCRATCH FILE JS
C     BY SUBROUTINE SIMK
C
      READ(JS,160)CARD
      IF(CARD(1).NE.HOUTP)GO TO 1320
 1050 CONTINUE
```

Figure 14. Subroutine NAMEL Program Listing (Continued)

```
      READ(JS,1060)II,K,I
 1060 FORMAT(3I2)
      IF(II.EQ.-1)GO TO 1110
      NNO(II)=II
      ENCODE(4,420,VNO(II,1))HRP,II
      VNO(II,2)=HP
      DO 1080 J=1,10
 1080 DESO(II,J)=DESOO(I,J,K)
      DO 1100 J=1,4
 1100 UNITO(II,J)=UNITOO(I,J,K)
      GO TO 1050
 1110 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(7,4HNAME,4HL    ,0,0,IW)
C
C     READ NAME LIST DATA FOR INPUTS OBTAINABLE FROM
C     INTERCONNECTION EQUATIONS WRITTEN ON SCRATCH FILE JS
C     BY SUBROUTINE SIMK
C
      READ(JS,160)CARD
      IF(CARD(1).NE.HINPU)GO TO 1320
 1120 CONTINUE
      READ(JS,1060)II,K,I
      IF(II.EQ.-1)GO TO 1170
      NNI(II)=II
      ENCODE(4,420,VNI(II,1))HUP,II
      VNI(II,2)=HP
      DO 1140 J=1,10
 1140 DESI(II,J)=DESII(I,J,K)
      DO 1160 J=1,4
 1160 UNITI(II,J)=UNITII(I,J,K)
      GO TO 1120
 1170 CONTINUE
      READ(JS,160)CARD
      IF(CARD(1).NE.HEND)GO TO 1320
C
C     READ NAME LIST DATA FROM CARDS
C
 1220 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(8,4HNAME,4HL    ,0,0,IW)
      READ(IR,160)CARD
  160 FORMAT(20A4)
      IF(CARD(1).EQ.HEND)GO TO 1340
      IF(CARD(1).EQ.HSTAT)GO TO 1240
      IF(CARD(1).EQ.HOUTP)GO TO 1260
      IF(CARD(1).EQ.HINPU)GO TO 1300
      GO TO 200
C
C     READ NAME LIST DATA FOR STATES
C
 1240 CONTINUE
      READ(IR,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4)
  280 FORMAT(I2,6X,2A4,4X,10A4,4X,4A4)
      IF(NNNN.EQ.-1)GO TO 1220
      NNS(NNNN)=NNNN
      DO 1245 J=1,2
 1245 VNS(NNNN,J)=VN(J)
      DO 1250 J=1,10
 1250 DESS(NNNN,J)=DES(J)
      DO 1255 J=1,4
 1255 UNITS(NNNN,J)=UNIT(J)
      GO TO 1240
C
C     READ NAME LIST DATA FOR OUTPUTS
C
 1260 CONTINUE
      READ(IR,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4)
```

Figure 14. Subroutine NAMEL Program Listing (Continued)

```
      IF(NNNN.EQ.-1)GO TO 1220
      NNO(NNNN)=NNNN
      DO 1265 J=1,2
 1265 VNO(NNNN,J)=VN(J)
      DO 1270 J=1,10
 1270 DESO(NNNN,J)=DES(J)
      DO 1275 J=1,4
 1275 UNITO(NNNN,J)=UNIT(J)
      GO TO 1260
C
C     READ NAME LIST DATA FOR INPUTS
C
 1300 CONTINUE
      READ(IR,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4)
      IF(NNNN.EQ.-1)GO TO 1220
      NNI(NNNN)=NNNN
      DO 1285 J=1,2
 1285 VNI(NNNN,J)=VN(J)
      DO 1290 J=1,10
 1290 DESI(NNNN,J)=DES(J)
      DO 1295 J=1,4
 1295 UNITI(NNNN,J)=UNIT(J)
      GO TO 1300
 1340 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(9,4HNAME,4HL    ,0,0,IW)
C
C     PRINT HEADING AND NAME LIST DATA
C
      IF(IPRINT.LT.2)GO TO 1540
      CALL HPR(HEAD,IW)
      WRITE(9,1360)NX,NR,NU
 1360 FORMAT(//,1X,18HNUMBER OF STATES =,I2,//,1X,
     118HNUMBER OF OUTPUTS=,I2,//,1X,18HNUMBER OF INPUTS =,I2,//)
      WRITE(IW,1380)
 1380 FORMAT(//,20X,23H*** NAME LIST TABLE ***,/)
      WRITE(IW,1400)
 1400 FORMAT(/,1X,8HVARIABLE,6H NAME ,6X,13H DESCRIPTION ,
     131X,6H UNIT ,/)
      IF(IPRINT.EQ.6)CALL DEBUG(10,4HNAME,4HL    ,0,0,IW)
C
C     PRINT NAME LIST DATA FOR STATES
C
      WRITE(IW,1460)
 1460 FORMAT(/,1X,6HSTATE ,/)
      WRITE(IW,1480)(NNS(I),(VNS(I,J),J=1,2),(DESS(I,J),J=1,10),
     1              (UNITS(I,J),J=1,4),I=1,NX)
 1480 FORMAT(1X,I2,6X,2A4,4X,10A4,4X,4A4)
C
C     PRINT NAME LIST DATA FOR OUTPUTS
C
      WRITE(IW,1500)
 1500 FORMAT(/,1X,6HOUTPUT,/)
      WRITE(IW,1480)(NNO(I),(VNO(I,J),J=1,2),(DESO(I,J),J=1,10),
     1              (UNITO(I,J),J=1,4),I=1,NR)
C
C     PRINT NAME LIST DATA FOR INPUTS
C
      WRITE(IW,1520)
 1520 FORMAT(/,1X,6HINPUT ,/)
      WRITE(IW,1480)(NNI(I),(VNI(I,J),J=1,2),(DESI(I,J),J=1,10),
     1              (UNITI(I,J),J=1,4),I=1,NU)
 1540 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(11,4HNAME,4HL    ,0,0,IW)
C
C     WRITE NAME LIST DATA ON DISK FILE
C
```

Figure 14. Subroutine NAMEL Program Listing (Continued)

```
      CALL FILE(JN,INSERT,HEAD)                                         NAMEL395
      WRITE(JN)NX,NR,NU,                                                NAMEL396
     1         (NNS(I),(VNS(I,J),J=1,2),                                NAMEL397
     2         (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NX),           NAMEL398
     3         (NNO(I),(VNO(I,J),J=1,2),                                NAMEL399
     4         (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NR),           NAMEL400
     5         (NNI(I),(VNI(I,J),J=1,2),                                NAMEL401
     6         (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NU)            NAMEL402
      CALL FILE(JN,INSERT,MARK)                                         NAMEL403
      IF(IPRINT.EQ.6)CALL DEBUG(12,4HNAME,4HL    ,0,0,IW)               NAMEL404
      RETURN                                                            NAMEL405
C                                                                       NAMEL406
C     PRINT ERROR MESSEGE                                               NAMEL407
C                                                                       NAMEL408
  200 CONTINUE                                                          NAMEL409
      WRITE(IW,220)                                                     NAMEL410
  220 FORMAT(1H1,//,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)        NAMEL411
      STOP 111                                                          NAMEL412
 1320 CONTINUE                                                          NAMEL413
      WRITE(IW,1330)                                                    NAMEL414
 1330 FORMAT(1H1,//,1X,30HERROR IN DATA PROVIDED BY SIMK)               NAMEL415
      STOP 111                                                          NAMEL416
      END                                                               NAMEL417
```

Figure 14. Subroutine NAMEL Program Listing (Concluded)

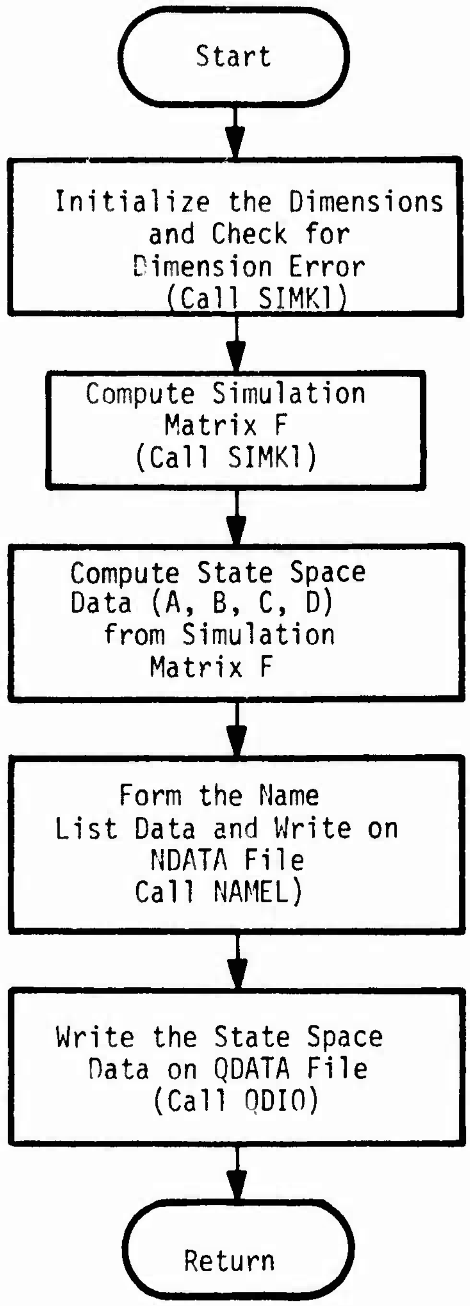

Figure 15. Subroutine STAMK1 Flow Chart

```
      SUBROUTINE STAMK1(V,W,F,U,A,B,C,D,
     1NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,
     2MAXN,MAXM,NXM,NRM,NUM,NYM,MB,MS1,MS2,MS3,MS4,NB)
C
C     PURPOSE - TO OBTAIN STATE SPACE MODEL OF THE VEHICLE
C     DESCRIBED BY SIMULATOR DECK DATA FROM LSA PROGRAM
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C         DEBUG
C         DEPRMS
C         MPRS
C         QDIO
C         TDINVR
C         DEPRM
C         HPP
C         NAMEL
C         SIMK1
C
C     ARGUMENTS LIST
C         V                        V ARRAY FOR COMPUTING SIMULATION MATRIX
C         W                        W ARRAY FOR COMPUTING SIMULATION MATRIX
C         F                        SIMULATION MATRIX
C         U                        ARRAY FOR EXTERNAL INPUTS
C         A           IN/OUT       STATE TRANSITION MATRIX
C         B           IN/OUT       CONTROL INPUT MATRIX
C         C           IN/OUT       STATE OUTPUT MATRIX
C         D           IN/OUT       CONTROL OUTPUT MATRIX
C         NNS         IN/OUT       NUMBER ARRAY FOR STATE
C         VNS         IN/OUT       VARIABLE NAME ARRAY FOR STATE
C         DESS        IN/OUT       DESCRIPTION ARRAY FOR STATE
C         UNITS       IN/OUT       UNIT ARRAY FOR STATE
C         NNO         IN/OUT       NUMBER ARRAY FOR OUTPUT
C         VNO         IN/OUT       VARIABLE NAME ARRAY FOR OUTPUT
C         DESO        IN/OUT       DESCRIPTION ARRAY FOR OUTPUT
C         UNITO       IN/OUT       UNIT ARRAY FOR OUTPUT
C         NNI         IN/OUT       NUMBER ARRAY FOR INPUT
C         VNI         IN/OUT       VARIABLE NAME ARRAY FOR INPUT
C         DESI        IN/OUT       DESCRIPTION ARRAY FOR INPUT
C         UNITI       IN/OUT       UNIT ARRAY FOR INPUT
C         MAXN        INPUT        MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C         MAXM        INPUT        MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C         NXM         INPUT        MAXIMUM NUMBER OF STATES
C         NRM         INPUT        MAXIMUM NUMBER OF OUTPUTS
C         NUM         INPUT        MAXIMUM NUMBER OF INPUTS
C         NYM         INPUT        MAXIMUM DIMENSION FOR INTERCONN EQUATIONS
C         MB          INPUT        MAXIMUM NO OF SUBSYSTEMS FOR COMBINING
C         MS1         INPUT        MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C         MS2         INPUT        MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C         MS3         INPUT        MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C         MS4         INPUT        MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C         NB          INPUT        MAXIMUM SYSTEM NO - IMPLICIT MODEL
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
      DIMENSION U(NUM)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
```

Figure 16. Subroutine STAMK1 Program Listing

```
      COMMON /SC1/ S1(1)                                                STAMK165
C     DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB)                       STAMK166
C     DIMENSION DESOO(NRM,10,MB),UNITOO(NRM,4,MB)                       STAMK167
C     DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)                       STAMK168
C     DIMENSION NXX(MB),NRR(MB),NUU(MB)                                 STAMK169
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSTAM,4HK1  ,1,0,IW)                 STAMK170
      L1=1 $ L2=L1+NXM*MB*10 $ L3=L2+NXM*MB*4 $ L4=L3+NRM*MB*10         STAMK171
      L5=L4+NRM*MB*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MB*4                 STAMK172
      L8=L7+MB $ L9=L8+MB $ L10=L9+MB                                   STAMK173
      IF(L10.GT.MS1)                                                    STAMK174
     1CALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,1,0,4HSTAM,4HK1  ,IW)  STAMK175
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HSTAM,4HK1  ,1,0,IW)                 STAMK176
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0                     STAMK177
      NXA=0 $ NRA=0 $ NUA=0                                             STAMK178
      EPSF=1.0E-30 $ T=0.0 $ NFLAG=0                                    STAMK179
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))CALL HPR(HEAD,IW)               STAMK180
C                                                                       STAMK181
C     INITIALIZING CALL TO SUBROUTINE STMK1                             STAMK182
C                                                                       STAMK183
      INIT=                                                             STAMK184
      NX=0 $ NY=0 $ NP=0 $ NU=0                                         STAMK185
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                             STAMK186
      CALL STMK1(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),             STAMK187
     1NX,NY,NR,NU,INIT,T,MS1,MS2,MS3,MS4)                               STAMK188
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HSTAM,4HK1  ,1,0,IW)                 STAMK189
C                                                                       STAMK190
C     CHECK FOR DIMENSION ERROR                                         STAMK191
C                                                                       STAMK192
      INIT = 1                                                          STAMK193
      M=2*NX+NY+NU                                                      STAMK194
      N=NX+NY+NR                                                        STAMK195
      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))      STAMK196
     1CALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,1,0,4HSTAM,4HK1  ,IW)     STAMK197
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                             STAMK198
      DO 101 J=1,M                                                      STAMK199
101   W(J)=0.                                                           STAMK100
      DO 501 J=1,M                                                      STAMK101
      W(J)=1.                                                           STAMK102
      CALL STMK1(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),             STAMK103
     1NX,NY,NR,NU,INIT,T,MS1,MS2,MS3,MS4)                               STAMK104
      W(J)=0.0                                                          STAMK105
      DO 501 I=1,N                                                      STAMK106
501   F(I,J)=V(I)                                                       STAMK107
C                                                                       STAMK108
C     COMPUTE THE SIMULATION MATRIX                                     STAMK109
C                                                                       STAMK110
      NV=NX+NY                                                          STAMK111
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIM )                STAMK112
      DO 51 I=1,NV                                                      STAMK113
      DO 52 J=1,NV                                                      STAMK114
   52 F(I,J)=-F(I,J)                                                    STAMK115
   51 F(I,I)=F(I,I)+1.                                                  STAMK116
      CALL TDINVR(ISOL,IDSOL,NV,-M,F,MAXN,KDUM,DET)                     STAMK117
      IB=NV+1                                                           STAMK118
      IE=NV+NR                                                          STAMK119
      JB=IB                                                             STAMK120
      JE=M                                                              STAMK121
      DO 53 I=IB,IE                                                     STAMK122
      DO 53 J=JB,JE                                                     STAMK123
      DO 53 K=1,NV                                                      STAMK124
   53 F(I,J)=F(I,J)+F(I,K)*F(K,J)                                       STAMK125
      DO 53  I=1,IE                                                     STAMK126
      DO 53  J=1,JE                                                     STAMK127
      IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0                              STAMK128
  530 CONTINUE                                                          STAMK129
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIMI)                STAMK130
```

Figure 16. Subroutine STAMK1 Program Listing (Continued)

```
C
C     FORM A,B,C,D MATRICES
C
      J1=NV+1
      J2=NV+NX
      J3=J1+NX
      J4=J2+NU
      I1=NV+1
      I2=NV+NR
      DO 6001 I=1,NX
      DO 6001 J=J1,J2
      JJ=J-J1+1
 6001 A(I,JJ)=F(I,J)
      DO 6002 I=1,NX
      DO 6002 J=J3,J4
      JJ=J-J3+1
 6002 B(I,JJ)=F(I,J)
      DO 6003 I=I1,I2
      II=I-I1+1
      DO 6003 J=J1,J2
      JJ=J-J1+1
 6003  C(II,JJ)=F(I,J)
      DO 6004 I=I1,I2
      II=I-I1+1
      DO 6004 J=J3,J4
      JJ=J-J3+1
 6004 D(II,JJ)=F(I,J)
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HSTAM,4HK1  ,1,0,IW)
C
C     READ AND UPDATE NAME LIST DATA
C
      KB=NMAX
      CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
     1DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
     2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HSTAM,4HK1  ,1,0,IW)
C
C     WRITE QUADRUPLE DATA ON FILE QDATA
C
      IQ=0
      MFLAG=2
      NXA=NX $ NRA=NR $ NUA=NU
      CALL QDIO(A,B,C,D,A,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,HEAD,MARK,
     2LOCATE,NULL,INSERT,MFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(6,4HSTAM,4HK1  ,1,0,IW)
      RETURN
      END
```

Figure 16. Subroutine STAMK1 Program Listing (Concluded)

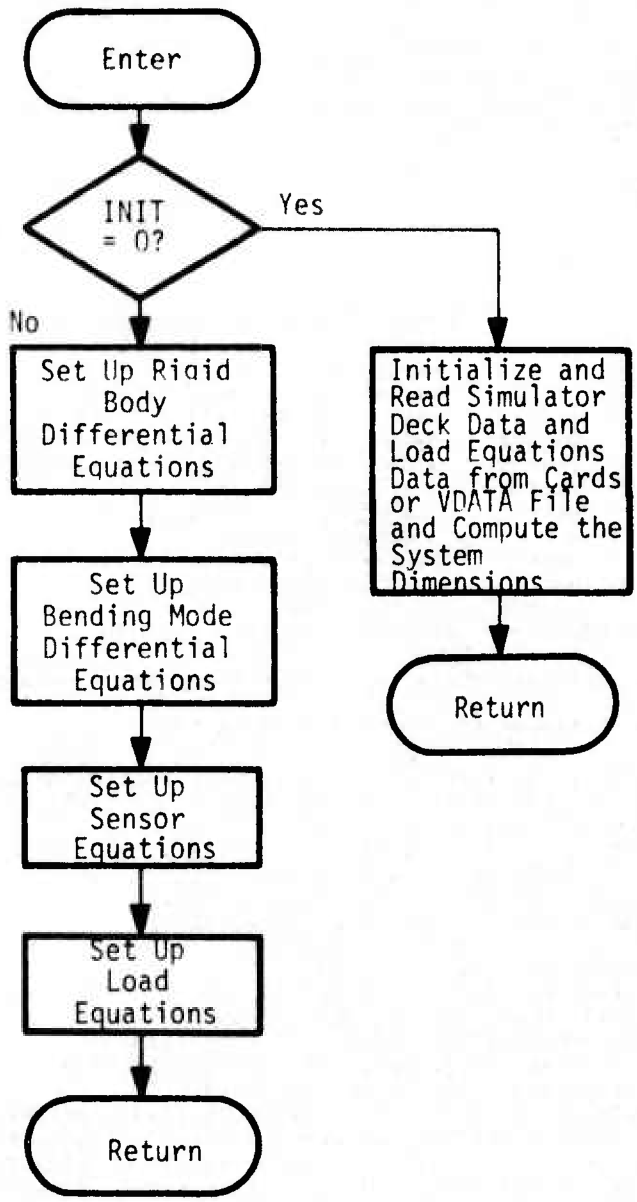

Figure 17. Subroutine SIMK1 Flow Chart

```
      SUBROUTINE SIMK1(XDOT,Y,X,U,XDOTL,YL,RL,NX,NY,NR,NU,INIT,T,
     1MS1,MS2,MS3,MS4)
C
C     PURPOSE - TO READ SIMULATOR MATRIX DATA FROM LSA AND
C     TO IMPLEMENT STANDARD LSA EQUATIONS
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - MAY 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        INPT1
C        MPRS1
C
C     ARGUMENTS LIST
C        XDOT                  ARRAY FOR STATE DERIVATIVES
C        Y                     ARRAY FOR Y EQUATIONS
C        X                     ARRAY FOR STATES
C        U                     ARRAY FOR EXTERNAL INPUTS
C        XDOTL       OUTPUT    ARRAY FOR DERIVATIVE OF STATE
C        YL          OUTPUT    ARRAY FOR Y EQUATION VARIABLES
C        RL          OUTPUT    ARRAY FOR EXTERNAL RESPONSE VARIABLES
C        NX          OUTPUT    NUMBER OF STATES
C        NY          OUTPUT    NUMBER OF Y EQUATIONS
C        NR          OUTPUT    NUMBER OF OUTPUTS
C        NU          OUTPUT    NUMBER OF INPUTS
C        INIT        INPUT     INITIAL MODE FLAG
C        T           OUTPUT    SAMPLE TIME
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION XDOT(NX),Y(NY),X(NX),U(NU),XDOTL(NX),YL(NY),RL(NR)
C
C     DIMENSION STATEMENT FOR THE MATRIX DATA FROM LSA
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      REAL LVP0,LVP1,LR0,LR1,LUF0,LUE1,LUE2,LDELS0,LDELS1,LDELS2
      REAL LUG0,LUG1,LVG0,LVG1,LWG0,LWG1
      COMMON /SC2/ VPVP0(6,6),VPVP1(6,6),VPR0(6,3),VPR1(6,3)
     1,VPUF0(6,30),VPUF1(6,30),VPUE2(6,30),VPDELS0(6,3)
     2,VPDELS1(6,3),VPDELS2(6,3),VPUG0(6,3),VPUG1(6,3)
     3,VPVG0(6,3),VPVG1(6,3),VPWG0(6,3),VPWG1(6,3)
     4,RVP0(3,6),RVP1(3,6),RR0(3,3),RR1(3,3)
     5,UFVP0(30,6),UFVP1(30,6),UER0(30,3)
     6,UFR1(30,3),UFUE0(30,30),UEUE1(30,30),UEUF2(30,30)
     7,UEDELS0(30,3),UEDELS1(30,3),UEDELS2(30,3)
     8,UEUG0(30,3),UEUG1(30,3),UEVG0(30,3),UEVG1(30,3)
     9,UEWG0(30,3),UFWG1(30,3),TVP0(9,6),TVP1(9,6),TR0(9,3)
     A,TR1(9,3),TUF0(9,30),TUE1(9,30),TUF2(9,30)
     B,TDELS0(9,3),TDELS1(9,3),TDELS2(9,3),TUG0(9,3),TUG1(9,3)
     C,TVG0(9,3),TVG1(9,3),TWG0(9,3),TWG1(9,3),LVP0(15,6),LVP1(15,6)
     D,LR0(15,3),LR1(15,3),LUF0(15,30),LUE1(15,30),LUE2(15,30),
     E,LDELS0(15,3),LDELS1(15,3),LDELS2(15,3),LUG0(15,3),LUG1(15,3)
     F,LVG0(15,3),LVG1(15,3),LWG0(15,3),LWG1(15,3)
     G,BANDING(1,3),UNITY(30,30)
      DIMENSION JHEAD(120),IDRM(120),IDCM(120)
      DIMENSION ICARD(8),IHEAD(8)
      DIMENSION SC(1)
      EQUIVALENCE (SC(1),VPVP0(1,1))
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSIMK,4H1    ,1,0,IW)
      IF(INIT.NE.0) GO TO 150
C
C     INITIALIZE AND SET MAX DIMENSIONS FOR SIMULATOR MATRIX DATA
C
      IEND=10HEND
      NXVP=0 $ NXR=0 $ NXUE=0
```

Figure 18. Subroutine SIMK1 Program Listing

```
      NUC1=0 $ NUC2=0 $ NUC3=0
      NUG0=0 $ NUG1=0 $ NUGS0=0 $ NUGS1=0
      NVG0=0 $ NVG1=0 $ NVGS0=0 $ NVGS1=0
      NWG0=0 $ NWG1=0 $ NWGS0=0 $ NWGS1=0
      NRT=0 $ NRL=0 $ NB=1 $ NL=3
      NXVPM=6 $ NXRM=3 $ NXUEM=30
      NUCM=3 $ NUGM=3
      NRTM=9 $ NRLM=15 $ NBM=1 $ NLM=3
C
C     DEFINE NAMES FOR SIMULATOR MATRIX DATA
C
C     RIGID BODY VELOCITY COEFT MATRIX NAMES
C
      JHEAD(1)=10HVP/VP0      $ JHEAD(2)=10HVP/VP1
      JHEAD(3)=10HVP/R0       $ JHEAD(4)=10HVP/R1
      JHEAD(5)=10HVP/UE0      $ JHEAD(6)=10HVP/UE1
      JHEAD(7)=10HVP/UE2      $ JHEAD(8)=10HVP/DELS0
      JHEAD(9)=10HVP/DELS1    $ JHEAD(10)=10HVP/DELS2
      JHEAD(11)=10HVP/UG0      $ JHEAD(12)=10HVP/UG1
      JHEAD(13)=10HVP/VG0      $ JHEAD(14)=10HVP/VG1
      JHEAD(15)=10HVP/WG0      $ JHEAD(16)=10HVP/WG1
      JHEAD(17)=10HVP/UGS0     $ JHEAD(18)=10HVP/UGS1
      JHEAD(19)=10HVP/VGS0     $ JHEAD(20)=10HVP/VGS1
      JHEAD(21)=10HVP/WGS0     $ JHEAD(22)=10HVP/WGS1
C
C     RIGID BODY ATTITUDE COEFT MATRIX NAMES
C
      JHEAD(23)=10HR/VP0       $ JHEAD(24)=10HR/VP1
      JHEAD(25)=10HR/R0        $ JHEAD(26)=10HR/R1
C
C     BENDING MODE COEFT MATRIX NAMES
C
      JHEAD(45)=10HUE/VP0      $ JHEAD(46)=10HUE/VP1
      JHEAD(47)=10HUE/R0       $ JHEAD(48)=10HUE/R1
      JHEAD(49)=10HUE/UE0      $ JHEAD(50)=10HUE/UE1
      JHEAD(51)=10HUE/UE2      $ JHEAD(52)=10HUE/DELS0
      JHEAD(53)=10HUE/DELS1    $ JHEAD(54)=10HUE/DELS2
      JHEAD(55)=10HUE/UG0      $ JHEAD(56)=10HUE/UG1
      JHEAD(57)=10HUE/VG0      $ JHEAD(58)=10HUE/VG1
      JHEAD(59)=10HUE/WG0      $ JHEAD(60)=10HUE/WG1
      JHEAD(61)=10HUE/UGS0     $ JHEAD(62)=10HUE/UGS1
      JHEAD(63)=10HUE/VGS0     $ JHEAD(64)=10HUE/VGS1
      JHEAD(65)=10HUE/WGS0     $ JHEAD(66)=10HUE/WGS1
C
C     SENSOR COEFT MATRIX NAMES
C
      JHEAD(67)=10HT/VP0       $ JHEAD(68)=10HT/VP1
      JHEAD(69)=10HT/R0        $ JHEAD(70)=10HT/R1
      JHEAD(71)=10HT/UE0       $ JHEAD(72)=10HT/UE1
      JHEAD(73)=10HT/UE2       $ JHEAD(74)=10HT/DELS0
      JHEAD(75)=10HT/DELS1     $ JHEAD(76)=10HT/DELS2
      JHEAD(77)=10HT/UG0       $ JHEAD(78)=10HT/UG1
      JHEAD(79)=10HT/VG0       $ JHEAD(80)=10HT/VG1
      JHEAD(81)=10HT/WG0       $ JHEAD(82)=10HT/WG1
      JHEAD(83)=10HT/UGS0      $ JHEAD(84)=10HT/UGS1
      JHEAD(85)=10HT/VGS0      $ JHEAD(86)=10HT/VGS1
      JHEAD(87)=10HT/WGS0      $ JHEAD(88)=10HT/WGS1
C
C     LOADS COEFT MATRIX NAMES
C
      JHEAD(89)=10HL/VP0       $ JHEAD(90)=10HL/VP1
      JHEAD(91)=10HL/R0        $ JHEAD(92)=10HL/R1
      JHEAD(93)=10HL/UE0       $ JHEAD(94)=10HL/UE1
      JHEAD(95)=10HL/UE2       $ JHEAD(96)=10HL/DELS0
      JHEAD(97)=10HL/DELS1     $ JHEAD(98)=10HL/DELS2
      JHEAD(99)=10HL/UG0       $ JHEAD(100)=10HL/UG1
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      JHEAD(101)=10HL/VG0       $  JHEAD(102)=10HL/VG1
      JHEAD(103)=10HL/WG0       $  JHEAD(104)=10HL/WG1
      JHEAD(105)=10HL/UGS0      $  JHEAD(106)=10HL/UGS1
      JHEAD(107)=10HL/VGS0      $  JHEAD(108)=10HL/VGS1
      JHEAD(109)=10HL/WGS0      $  JHEAD(110)=10HL/WGS1
C
C     MISCELLANEOUS MATRIX NAMES
C
      JHEAD(111)=10H(BANDING)   $  JHEAD(112)=10H*FINISHED*
C
C     SET UP MAX ROW AND COL DIMENSIONS FOR SIMULATOR MATRIX DATA
C
      DO 4 I=1,16
      I1=I
      I2=16+I
      I3=32+I
      I4=48+I
      I5=64+I
      IDRM(I1)=NXVPM $ IDRM(I2)=NXRM $ IDRM(I3)=NXUEM
      IDRM(I4)=NRTM $ IDRM(I5)=NRLM
    4 CONTINUE
      DO 6 I=1,5
      J=(16*I-16)
      IDCM(J+1)=NXVPM $ IDCM(J+2)=NXVPM
      IDCM(J+3)=NXRM  $ IDCM(J+4)=NXRM
      IDCM(J+5)=NXUFM $ IDCM(J+6)=NXUFM $ IDCM(J+7)=NXUFM
      IDCM(J+8)=NUCM  $ IDCM(J+9)=NUCM $ IDCM(J+10)=NUCM
      IDCM(J+11)=NUGM $ IDCM(J+12)=NUGM $ IDCM(J+13)=NUGM
      IDCM(J+14)=NUGM $ IDCM(J+15)=NUGM $ IDCM(J+16)=NUGM
    6 CONTINUE
      IDRM(81)=NRM $ IDCM(81)=NLM
      IDRM(82)=30 $ IDCM(82)=30
C
C     CHECK IF SCRATCH ARRAY SIZE IS SUFFICIENT
C
      N=0
      DO 8 I=1,20
    8 N=N+IDRM(I)*IDCM(I)
      DO 9 I=33,82
    9 N=N+IDRM(I)*IDCM(I)
      IF(N.GT.MS2)
     1CALL DFRRM(MS1,N,MS3,MS4,MS1,MS2,MS3,MS4,1,0,4HSIMK,4H1    ,IW)
      IF(IPRINT.EQ.6)CALL DFBUG(2,4HSIMK,4H1    ,1,0,IW)
C
C     INITIALIZE THE MEMORY WHERE SIMULATOR MATRIX DATA IS STORED
C
      DO 10 I=1,N
   10 SC(I)=0.0
C
C     READ LSA SIMULATOR DECK IDENTIFICATION CARD
C
   12 CONTINUE
      READ(IR,16)ICARD
   16 FORMAT(8A10)
      IF(ICARD(1).EQ.IEND)RETURN
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,22)
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,24)ICARD
   22 FORMAT(//,20X,27H*** LSA - FLEXSTAB DATA ***,//)
   24 FORMAT(//,1X,8A10,//)
      DO 28 I=1,8
   28 IHEAD(I)=ICARD(I)
      READ(IR,16)ICARD
      MHEAD=ICARD(1)
      DECODE(10,30,ICARD(2))NROW,NCOL
   30 FORMAT(2I5)
      IF(ICARD(1).NE.IEND)GO TO 52
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      JR=IR
      IR=4
C
C     LOCATE SIMULATOR DECK DATA
C
   32 CONTINUE
      READ(IR,16)ICARD
      IF(EOF(IR))36,44
   36 CONTINUE
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,40)IR
   40 FORMAT(1H1,//,1X,38HVEHICLE DATA CANNOT BE FOUND ON FILE= ,I2)
      STOP 111
C
C     READ MATRIX NAME CARD
C
   44 CONTINUE
      IF(IHEAD(1).NE.ICARD(1))GO TO 32
      READ(IR,16)ICARD
      MHEAD=ICARD(1)
      DECODE(10,30,ICARD(2))NROW,NCOL
      GO TO 52
   48 CONTINUE
      READ(IR,16)ICARD
      MHEAD=ICARD(1)
      DECODE(10,30,ICARD(2))NROW,NCOL
   52 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HSIMK,4H1    ,1,0,IW)
      DO 54 I=1,112
      IF(MHEAD.EQ.JHEAD(I))GO TO 58
   54 CONTINUE
C
C     PRINT ERROR MESSEGE
C
   55 CONTINUE
      WRITE(IW,56)
   56 FORMAT(1H1,//,1X,19HERROR IN INPUT DATA)
      STOP 111
C
C     READ AND PRINT LSA SIMULATOR DECK DATA
C
   58 CONTINUE
      IF(I.GE.112)GO TO 98
C
C     COMPUTE II FROM I SO THAT STEADY GUST COEFT MATRICES ARE
C     STORED IN THE SAME LOCATIONS AS THE GUST COEFT MATRICES
C
      II=I
      IF((I.GT.16).AND.(I.LE.26))II=I-6
      IF((I.GT.26).AND.(I.LE.44))GO TO 55
      IF((I.GT.44).AND.(I.LE.60))II=I-12
      IF((I.GT.60).AND.(I.LE.82))II=I-18
      IF((I.GT.82).AND.(I.LE.104))II=I-24
      IF((I.GT.104).AND.(I.LE.112))II=I-30
C
C     COMPUTE ARRAY START INDEX FOR SIMULATOR MATRIX DATA
C
      N=1
      IM1=II-1
      IF(II.EQ.33)IM1=20
      IF(IM1.EQ.0)GO TO 70
      JM1=IM1
      IF(IM1.GT.20)JM1=20
      DO 60 J=1,JM1
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
   60 N=N+IDRM(J)*IDCM(J)
      IF(IM1.LE.20)GO TO 70
      DO 65 J=33,IM1
   65 N=N+IDRM(J)*IDCM(J)
   70 CONTINUE
C
C     READ AND PRINT SIMULATOR MATRIX DATA
C
      NROWM=IDRM(II) $ NCOLM=IDCM(II)
      CALL INPT1(SC(N),NROWM,NCOLM,NROW,NCOL,IR)
      CALL MPRS1(SC(N),NROWM,NCOLM,NROW,NCOL,MHEAD)
C
C     COMPUTE STATE DIMENSION OF SIMULATOR MATRIX DATA READ
C
      IF((II.GT.00).AND.(II.LE.16))NXVP=NROW
      IF((II.GT.16).AND.(II.LE.32))NXR=NROW
      IF((II.GT.32).AND.(II.LE.48))NXUE=NROW
      IF((II.GT.48).AND.(II.LE.64))NRT=NROW
      IF((II.GT.64).AND.(II.LE.80))NRL=NROW
C
C     COMPUTE INPUT DIMENSION OF SIMULATOR MATRIX DATA READ
C
      DO 80 J=1,5
      JJ=(J-1)*22+I
      IF(JJ.LT.96)GO TO 80
      IF(JJ.GT.110)GO TO 80
      IF(JJ.EQ.96)NUC1=NCOL
      IF(JJ.EQ.97)NUC2=NCOL
      IF(JJ.EQ.98)NUC3=NCOL
      IF(JJ.EQ.99)NUG0=NCOL
      IF(JJ.EQ.100)NUG1=NCOL
      IF(JJ.EQ.101)NVG0=NCOL
      IF(JJ.EQ.102)NVG1=NCOL
      IF(JJ.EQ.103)NWG0=NCOL
      IF(JJ.EQ.104)NWG1=NCOL
      IF(JJ.EQ.105)NUGS0=NCOL
      IF(JJ.EQ.106)NUGS1=NCOL
      IF(JJ.EQ.107)NVGS0=NCOL
      IF(JJ.EQ.108)NVGS1=NCOL
      IF(JJ.EQ.109)NWGS0=NCOL
      IF(JJ.EQ.110)NWGS1=NCOL
   80 CONTINUE
      GO TO 48
C
C     PRINT THE LAST CARD READ FROM SIMULATOR MATRIX DATA
C
   98 CONTINUE
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,100)MHEAD
  100 FORMAT(//,10X,A10,/)
      IF(IPRINT.EQ.6)CALL DFBUG(4,4HSIMK,4H1    ,1,0,IW)
C
C     FORM THE UNITY MATRIX
C
      IF(NXUE.EQ.0)GO TO 134
      DO 130 I=1,NXUF
      DO 130 J=1,NXUF
      UNITY(I,J)=0.0
  130 UNITY(I,I)=1.0
  134 CONTINUE
C
C     CHECK FOR DIMENSION ERROR
C
      IF((NXVP.LE.NXVPM).AND.(NXR.LE.NXRM).AND.(NXUE.LE.NXUEM)
     1.AND.(NUC1.LE.NUCM).AND.(NUC2.LE.NUCM).AND.(NUC3.LE.NUCM)
     2.AND.(NUG0.LE.NUGM).AND.(NUG1.LE.NUGM)
     3.AND.(NVG0.LE.NUGM).AND.(NVG1.LE.NUGM)
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
     4.AND.(NWG0.LE.NUGM).AND.(NWG1.LE.NUGM)
     5.AND.(NUGS0.LE.NUGM).AND.(NUGS1.LE.NUGM)
     6.AND.(NVGS0.LE.NUGM).AND.(NVGS1.LE.NUGM)
     7.AND.(NWGS0.LE.NUGM).AND.(NWGS1.LE.NUGM)
     8.AND.(NRT.LE.NRTM).AND.(NRL.LE.NRLM).AND.(NB.LE.NBM).AND
     9.(NL.LE.NLM))GO TO 138
      WRITE(IW,136)
  136 FORMAT(1H1,//,1X,43HDIMENSION OF LSA DATA EXCEEDS THAT USED IN
     1,16HSUBROUTINE SIMK1)
      STOP 111
  138 CONTINUE
C
C     COMPUTE SYSTEM DIMENSIONS
C
      NX=NXVP+NXR+NXUE*2
      NU=NUC1+NUC2+NUC3+NUG0+NUG1+NVG0+NVG1+NWG0+NWG1
     1+NUGS0+NUGS1+NVGS0+NVGS1+NWGS0+NWGS1
      NR=NRT+NRL
      IF(IR.NE.4)GO TO 12
      IR=JR
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HSIMK,4H1    ,1,0,IW)
      RETURN
C
C     PRINT ERROR MESSEGE
C
  140 CONTINUE
      WRITE(IW,145)
  145 FORMAT(1H1,//,1X,35HDIMENSION ERROR IN SUBROUTINE SIMK1)
      STOP 111
  150 CONTINUE
C
C     DIFFERENTIAL EQUATIONS FOR RIGID BODY VELOCITIES
C
      IF(NXVP.LE.0)GO TO 264
      DO 260 I=1,NXVP
      XDOTL(I)=0.0
C
C     FROM RIGID BODY VELOCITIES
C
      DO 152 K=1,NXVP
  152 XDOTL(I)=XDOTL(I)+VPVP0(I,K)*X(K)+VPVP1(I,K)*XDOT(K)
      IF(NXR.LE.0)GO TO 160
C
C     FROM RIGID BODY ATTITUDES
C
      DO 156 K=1,NXR
      KK=NXVP+K
  156 XDOTL(I)=XDOTL(I)+VPR0(I,K)*X(KK)+VPR1(I,K)*XDOT(KK)
  160 CONTINUE
      IF(NXUE.LE.0)GO TO 168
C
C     FROM BENDING MODES
C
      DO 164 K=1,NXUE
      KK=NXVP+NXR+K
      KKK=NXVP+NXR+NXUE+K
  164 XDOTL(I)=XDOTL(I)+VPUE0(I,K)*X(KK)+VPUE1(I,K)*X(KKK)
     1+VPUE2(I,K)*XDOT(KKK)
  168 CONTINUE
C
C     FROM CONTROL SURFACE INPUTS
C
      IF(NUC1.LE.0)GO TO 184
      DO 172 K=1,NUC1
  172 XDOTL(I)=XDOTL(I)+VPDFLS0(I,K)*U(K)
      IF(NUC2.LE.0)GO TO 184
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      DO 176 K=1,NUC2
      KK=NUC1+K
  176 XDOTL(I)=XDOTL(I)+VPDFLS1(I,K)*U(KK)
      IF(NUC3.LE.0)GO TO 184
      DO 180 K=1,NUC3
      KK=NUC1+NUC2+K
  180 XDOTL(I)=XDOTL(I)+VPDFLS2(I,K)*U(KK)
  184 CONTINUE
      MU=NUC1+NUC2+NUC3
      II=I
C
C     FROM U-GUST INPUTS
C
      IF(NUG0.LE.0)GO TO 196
      DO 188 K=1,NUG0
      KK=MU+K
  188 XDOTL(II)=XDOTL(II)+VPUG0(I,K)*U(KK)
      IF(NUG1.LE.0)GO TO 196
      DO 192 K=1,NUG1
      KK=MU+NUG0+K
  192 XDOTL(II)=XDOTL(II)+VPUG1(I,K)*U(KK)
  196 CONTINUE
      MU=MU+NUG0+NUG1
C
C     FROM V-GUST INPUTS
C
      IF(NVG0.LE.0)GO TO 208
      DO 200 K=1,NVG0
      KK=MU+K
  200 XDOTL(II)=XDOTL(II)+VPVG0(I,K)*U(KK)
      IF(NVG1.LE.0)GO TO 208
      DO 204 K=1,NVG1
      KK=MU+NVG0+K
  204 XDOTL(II)=XDOTL(II)+VPVG1(I,K)*U(KK)
  208 CONTINUE
      MU=MU+NVG0+NVG1
C
C     FROM W-GUST INPUTS
C
      IF(NWG0.LE.0)GO TO 220
      DO 212 K=1,NWG0
      KK=MU+K
  212 XDOTL(II)=XDOTL(II)+VPWG0(I,K)*U(KK)
      IF(NWG1.LE.0)GO TO 220
      DO 216 K=1,NWG1
      KK=MU+NWG0+K
  216 XDOTL(II)=XDOTL(II)+VPWG1(I,K)*U(KK)
  220 CONTINUE
      MU=MU+NWG0+NWG1
C
C     FROM STEADY U-GUST INPUTS
C
      IF(NUGS0.LE.0)GO TO 232
      DO 224 K=1,NUGS0
      KK=MU+K
  224 XDOTL(II)=XDOTL(II)+VPUG0(I,K)*U(KK)
      IF(NUGS1.LE.0)GO TO 232
      DO 228 K=1,NUGS1
      KK=MU+NUGS0+K
  228 XDOTL(II)=XDOTL(II)+VPUG1(I,K)*U(KK)
  232 CONTINUE
      MU=MU+NUGS0+NUGS1
C
C     FROM STEADY V-GUST INPUTS
C
      IF(NVGS0.LE.0)GO TO 244
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      DO 236 K=1,NVGS0
      KK=MU+K
  236 XDOTL(II)=XDOTL(II)+VPVG0(I,K)*U(KK)
      IF(NVGS1.LE.0)GO TO 244
      DO 240 K=1,NVGS1
      KK=MU+NVGS0+K
  240 XDOTL(II)=XDOTL(II)+VPVG1(I,K)*U(KK)
  244 CONTINUE
      MU=MU+NVGS0+NVGS1
C
C     FROM STEADY W-GUST INPUTS
C
      IF(NWGS0.LE.0)GO TO 256
      DO 248 K=1,NWGS0
      KK=MU+K
  248 XDOTL(II)=XDOTL(II)+VPWG0(I,K)*U(KK)
      IF(NWGS1.LE.0)GO TO 256
      DO 252 K=1,NWGS1
      KK=MU+NWGS0+K
  252 XDOTL(II)=XDOTL(II)+VPWG1(I,K)*U(KK)
  256 CONTINUE
      MU=MU+NWGS0+NWGS1
  260 CONTINUE
  264 CONTINUE
C
C     DIFFERENTIAL EQUATIONS FOR RIGID BODY ATTITUDES
C
      IF(NXR.LE.0)GO TO 280
      DO 272 I=1,NXR
      II=NXVP+I
      XDOTL(II)=0.0
      IF(NXVP.LE.0)GO TO 272
C
C     FROM RIGID BODY VELOCITIES
C
      DO 268 K=1,NXVP
  268 XDOTL(II)=XDOTL(II)+RVP0(I,K)*X(K)+RVP1(I,K)*XDOT(K)
  272 CONTINUE
C
C     FROM RIGID BODY ATTITUDES
C
      DO 276 K=1,NXR
      KK=NXVP+K
  276 XDOTL(II)=XDOTL(II)+RR0(I,K)*X(KK)+RR1(I,K)*XDOT(KK)
  280 CONTINUE
C
C     DIFFERENTIAL EQUATIONS FOR BENDING MODE DISPLACEMENTS AND RATES
C
      IF(NXUE.LE.0)GO TO 396
      DO 284 I=1,NXUE
      II=NXVP+NXR+I
      XDOTL(II)=0.0
      DO 284 K=1,NXUE
      KK=NXVP+NXR+NXUE+K
  284 XDOTL(II)=XDOTL(II)+UNITY(I,K)*X(KK)
      DO 392 I=1,NXUE
      II=NXVP+NXR+NXUE+I
      XDOTL(II)=0.0
      IF(NXVP.LE.0)GO TO 292
C
C     FROM RIGID BODY VELOCITIES
C
      DO 288 K=1,NXVP
  288 XDOTL(II)=XDOTL(II)+UEVP0(I,K)*X(K)+UEVP1(I,K)*XDOT(K)
  292 CONTINUE
      IF(NXR.LE.0)GO TO 298
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
C
C     FROM RIGID BODY ATTITUDES
C
      DO 296 K=1,NXR
      KK=NXVP+K
  296 XDOTL(II)=XDOTL(II)+UFR0(I,K)*X(KK)+UFR1(I,K)*XDOT(KK)
  298 CONTINUE
C
C     FROM BENDING MODES
C
      DO 300 K=1,NXUF
      KK=NXVP+NXR+K
      KKK=NXVP+NXR+NXUF+K
  300 XDOTL(II)=XDOTL(II)+UFUF0(I,K)*X(KK)+UFUF1(I,K)*X(KKK)
     1+UFUF2(I,K)*XDOT(KKK)
C
C     FROM CONTROL SURFACE INPUTS
C
      IF(NUC1.LE.0)GO TO 316
      DO 304 K=1,NUC1
  304 XDOTL(II)=XDOTL(II)+UFDELS0(I,K)*U(K)
      IF(NUC2.LE.0)GO TO 316
      DO 308 K=1,NUC2
      KK=NUC1+K
  308 XDOTL(II)=XDOTL(II)+UFDELS1(I,K)*U(KK)
      IF(NUC3.LE.0)GO TO 316
      DO 312 K=1,NUC3
      KK=NUC1+NUC2+K
  312 XDOTL(II)=XDOTL(II)+UFDELS2(I,K)*U(KK)
  316 CONTINUE
      MU=NUC1+NUC2+NUC3
C
C     FROM U-GUST INPUTS
C
      IF(NUG0.LE.0)GO TO 328
      DO 320 K=1,NUG0
      KK=MU+K
  320 XDOTL(II)=XDOTL(II)+UFUG0(I,K)*U(KK)
      IF(NUG1.LE.0)GO TO 328
      DO 324 K=1,NUG1
      KK=MU+NUG0+K
  324 XDOTL(II)=XDOTL(II)+UFUG1(I,K)*U(KK)
  328 CONTINUE
      MU=MU+NUG0+NUG1
C
C     FROM V-GUST INPUTS
C
      IF(NVG0.LE.0)GO TO 340
      DO 332 K=1,NVG0
      KK=MU+K
  332 XDOTL(II)=XDOTL(II)+UFVG0(I,K)*U(KK)
      IF(NVG1.LE.0)GO TO 340
      DO 336 K=1,NVG1
      KK=MU+NVG0+K
  336 XDOTL(II)=XDOTL(II)+UFVG1(I,K)*U(KK)
  340 CONTINUE
      MU=MU+NVG0+NVG1
C
C     FROM W-GUST INPUTS
C
      IF(NWG0.LE.0)GO TO 352
      DO 344 K=1,NWG0
      KK=MU+K
  344 XDOTL(II)=XDOTL(II)+UFWG0(I,K)*U(KK)
      IF(NWG1.LE.0)GO TO 352
      DO 348 K=1,NWG1
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      KK=MU+NWG0+K
  348 XDOTL(II)=XDOTL(II)+UFWG1(I,K)*U(KK)
  352 CONTINUE
      MU=MU+NWG0+NWG1
C
C     FROM STEADY U-GUST INPUTS
C
      IF(NUGS0.LE.0)GO TO 364
      DO 356 K=1,NUGS0
      KK=MU+K
  356 XDOTL(II)=XDOTL(II)+UFUG0(I,K)*U(KK)
      IF(NUGS1.LE.0)GO TO 364
      DO 360 K=1,NUGS1
      KK=MU+NUGS0+K
  360 XDOTL(II)=XDOTL(II)+UFUG1(I,K)*U(KK)
  364 CONTINUE
      MU=MU+NUGS0+NUGS1
C
C     FROM STEADY V-GUST INPUTS
C
      IF(NVGS0.LE.0)GO TO 376
      DO 368 K=1,NVGS0
      KK=MU+K
  368 XDOTL(II)=XDOTL(II)+UFVG0(I,K)*U(KK)
      IF(NVGS1.LE.0)GO TO 376
      DO 372 K=1,NVGS1
      KK=MU+NVGS0+K
  372 XDOTL(II)=XDOTL(II)+UFVG1(I,K)*U(KK)
  376 CONTINUE
      MU=MU+NVGS0+NVGS1
C
C     FROM STEADY W-GUST INPUTS
C
      IF(NWGS0.LE.0)GO TO 388
      DO 380 K=1,NWGS0
      KK=MU+K
  380 XDOTL(II)=XDOTL(II)+UFWG0(I,K)*U(KK)
      IF(NWGS1.LE.0)GO TO 388
      DO 384 K=1,NWGS1
      KK=MU+NWGS0+K
  384 XDOTL(II)=XDOTL(II)+UFWG1(I,K)*U(KK)
  388 CONTINUE
      MU=MU+NWGS0+NWGS1
  392 CONTINUE
  396 CONTINUE
C
C     SENSOR EQUATIONS
C
      IF(NRT.LE.0)GO TO 516
      DO 512 I=1,NRT
      RL(I)=0.0
      IF(NXVP.LE.0)GO TO 404
C
C     FROM RIGID BODY VELOCITIES
C
      DO 400 K=1,NXVP
  400 RL(I)=RL(I)+TVP0(I,K)*X(K)+TVP1(I,K)*XDOT(K)
  404 CONTINUE
      IF(NXR.LE.0)GO TO 412
C
C     FROM RIGID BODY ATTITUDES
C
      DO 408 K=1,NXR
      KK=NXVP+K
  408 RL(I)=RL(I)+TR0(I,K)*X(KK)+TR1(I,K)*XDOT(KK)
  412 CONTINUE
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      IF(NXUF.LE.0)GO TO 420
C
C     FROM BENDING MODES
C
      DO 416 K=1,NXUF
      KK=NXVP+NXR+K
      KKK=NXVP+NXR+NXUF+K
  416 RL(I)=RL(I)+TUF0(I,K)*X(KK)+TUF1(I,K)*X(KKK)+TUF2(I,K)*XDOT(KKK)
  420 CONTINUE
C
C     FROM CONTROL SURFACE INPUTS
C
      IF(NUC1.LE.0)GO TO 436
      DO 424 K=1,NUC1
  424 RL(I)=RL(I)+TDFLS0(I,K)*U(K)
      IF(NUC2.LE.0)GO TO 436
      DO 428 K=1,NUC2
      KK=NUC1+K
  428 RL(I)=RL(I)+TDFLS1(I,K)*U(KK)
      IF(NUC3.LE.0)GO TO 436
      DO 432 K=1,NUC3
      KK=NUC1+NUC2+K
  432 RL(I)=RL(I)+TDFLS2(I,K)*U(KK)
  436 CONTINUE
      MU=NUC1+NUC2+NUC3
C
C     FROM U-GUST INPUTS
C
      IF(NUG0.LE.0)GO TO 448
      DO 440 K=1,NUG0
      KK=MU+K
  440 RL(I)=RL(I)+TUG0(I,K)*U(KK)
      IF(NUG1.LE.0)GO TO 448
      DO 444 K=1,NUG1
      KK=MU+NUG0+K
  444 RL(I)=RL(I)+TUG1(I,K)*U(KK)
  448 CONTINUE
      MU=MU+NUG0+NUG1
C
C     FROM V-GUST INPUTS
C
      IF(NVG0.LE.0)GO TO 460
      DO 452 K=1,NVG0
      KK=MU+K
  452 RL(I)=RL(I)+TVG0(I,K)*U(KK)
      IF(NVG1.LE.0)GO TO 460
      DO 456 K=1,NVG1
      KK=MU+NVG0+K
  456 RL(I)=RL(I)+TVG1(I,K)*U(KK)
  460 CONTINUE
      MU=MU+NVG0+NVG1
C
C     FROM W-GUST INPUTS
C
      IF(NWG0.LE.0)GO TO 472
      DO 464 K=1,NWG0
      KK=MU+K
  464 RL(I)=RL(I)+TWG0(I,K)*U(KK)
      IF(NWG1.LE.0)GO TO 472
      DO 468 K=1,NWG1
      KK=MU+NWG0+K
  468 RL(I)=RL(I)+TWG1(I,K)*U(KK)
  472 CONTINUE
      MU=MU+NWG0+NWG1
C
C     FROM STEADY U-GUST INPUTS
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      IF(NUGS0.LE.0)GO TO 484
      DO 476 K=1,NUGS0
      KK=MU+K
  476 RL(I)=RL(I)+TUG0(I,K)*U(KK)
      IF(NUGS1.LE.0)GO TO 484
      DO 480 K=1,NUGS1
      KK=MU+NUGS0+K
  480 RL(I)=RL(I)+TUG1(I,K)*U(KK)
  484 CONTINUE
      MU=MU+NUGS0+NUGS1
C
C     FROM STEADY V-GUST INPUTS
C
      IF(NVGS0.LE.0)GO TO 496
      DO 488 K=1,NVGS0
      KK=MU+K
  488 RL(I)=RL(I)+TVG0(I,K)*U(KK)
      IF(NVGS1.LE.0)GO TO 496
      DO 492 K=1,NVGS1
      KK=MU+NVGS0+K
  492 RL(I)=RL(I)+TVG1(I,K)*U(KK)
  496 CONTINUE
      MU=MU+NVGS0+NVGS1
C
C     FROM STEADY W-GUST INPUTS
C
      IF(NWGS0.LE.0)GO TO 508
      DO 500 K=1,NWGS0
      KK=MU+K
  500 RL(I)=RL(I)+TWG0(I,K)*U(KK)
      IF(NWGS1.LE.0)GO TO 508
      DO 504 K=1,NWGS1
      KK=MU+NWGS0+K
  504 RL(I)=RL(I)+TWG1(I,K)*U(KK)
  508 CONTINUE
      MU=MU+NWGS0+NWGS1
  512 CONTINUE
  516 CONTINUE
C
C     LOAD EQUATIONS
C
      IF(NRL.LE.0)GO TO 716
      DO 712 I=1,NRL
      J=NRT+I
      RL(J)=0.0
      IF(NXVP.LE.0)GO TO 604
C
C     FROM RIGID BODY VELOCITIES
C
      DO 600 K=1,NXVP
  600 RL(J)=RL(J)+LVP0(I,K)*X(K)+LVP1(I,K)*XDOT(K)
  604 CONTINUE
      IF(NXR.LE.0)GO TO 612
C
C     FROM RIGID BODY ATTITUDES
C
      DO 608 K=1,NXR
      KK=NXVP+K
  608 RL(J)=RL(J)+LR0(I,K)*X(KK)+LR1(I,K)*XDOT(KK)
  612 CONTINUE
      IF(NXUE.LE.0)GO TO 620
C
C     FROM BENDING MODES
C
      DO 616 K=1,NXUE
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      KK=NXVP+NXR+K
      KKK=NXVP+NXR+NXUF+K
  616 RL(J)=RL(J)+LUF0(I,K)*X(KK)+LUF1(I,K)*X(KKK)+LUF2(I,K)*XDOT(KKK)
  620 CONTINUE
C
C     FROM CONTROL SURFACE INPUTS
C
      IF(NUC1.LE.0)GO TO 636
      DO 624 K=1,NUC1
  624 RL(J)=RL(J)+LDFLS0(I,K)*U(K)
      IF(NUC2.LE.0)GO TO 636
      DO 628 K=1,NUC2
      KK=NUC1+K
  628 RL(J)=RL(J)+LDFLS1(I,K)*U(KK)
      IF(NUC3.LE.0)GO TO 636
      DO 632 K=1,NUC3
      KK=NUC1+NUC2+K
  632 RL(J)=RL(J)+LDFLS2(I,K)*U(KK)
  636 CONTINUE
      MU=NUC1+NUC2+NUC3
C
C     FROM U-GUST INPUTS
C
      IF(NUG0.LE.0)GO TO 648
      DO 640 K=1,NUG0
      KK=MU+K
  640 RL(J)=RL(J)+LUG0(I,K)*U(KK)
      IF(NUG1.LE.0)GO TO 648
      DO 644 K=1,NUG1
      KK=MU+NUG0+K
  644 RL(J)=RL(J)+LUG1(I,K)*U(KK)
  648 CONTINUE
      MU=MU+NUG0+NUG1
C
C     FROM V-GUST INPUTS
C
      IF(NVG0.LE.0)GO TO 660
      DO 652 K=1,NVG0
      KK=MU+K
  652 RL(J)=RL(J)+LVG0(I,K)*U(KK)
      IF(NVG1.LE.0)GO TO 660
      DO 656 K=1,NVG1
      KK=MU+NVG0+K
  656 RL(J)=RL(J)+LVG1(I,K)*U(KK)
  660 CONTINUE
      MU=MU+NVG0+NVG1
C
C     FROM W-GUST INPUTS
C
      IF(NWG0.LE.0)GO TO 672
      DO 664 K=1,NWG0
      KK=MU+K
  664 RL(J)=RL(J)+LWG0(I,K)*U(KK)
      IF(NWG1.LE.0)GO TO 672
      DO 668 K=1,NWG1
      KK=MU+NWG0+K
  668 RL(J)=RL(J)+LWG1(I,K)*U(KK)
  672 CONTINUE
      MU=MU+NWG0+NWG1
C
C     FROM STEADY U-GUST INPUTS
C
      IF(NUGS0.LE.0)GO TO 684
      DO 676 K=1,NUGS0
      KK=MU+K
  676 RL(J)=RL(J)+LUG0(I,K)*U(KK)
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

```
      IF(NUGS1.LE.0)GO TO 684
      DO 680 K=1,NUGS1
      KK=MU+NUGS0+K
  680 RL(J)=RL(J)+LUG1(I,K)*U(KK)
  684 CONTINUE
      MU=MU+NUGS0+NUGS1
C
C     FROM STEADY V-GUST INPUTS
C
      IF(NVGS0.LE.0)GO TO 696
      DO 688 K=1,NVGS0
      KK=MU+K
  688 RL(J)=RL(J)+LVG0(I,K)*U(KK)
      IF(NVGS1.LE.0)GO TO 696
      DO 692 K=1,NVGS1
      KK=MU+NVGS0+K
  692 RL(J)=RL(J)+LVG1(I,K)*U(KK)
  696 CONTINUE
      MU=MU+NVGS0+NVGS1
C
C     FROM STEADY W-GUST INPUTS
C
      IF(NWGS0.LE.0)GO TO 708
      DO 700 K=1,NWGS0
      KK=MU+K
  700 RL(J)=RL(J)+LWG0(I,K)*U(KK)
      IF(NWGS1.LE.0)GO TO 708
      DO 704 K=1,NWGS1
      KK=MU+NWGS0+K
  704 RL(J)=RL(J)+LWG1(I,K)*U(KK)
  708 CONTINUE
      MU=MU+NWGS0+NWGS1
  712 CONTINUE
  716 CONTINUE
      RETURN
      END
```

Figure 18. Subroutine SIMK1 Program Listing (Concluded)

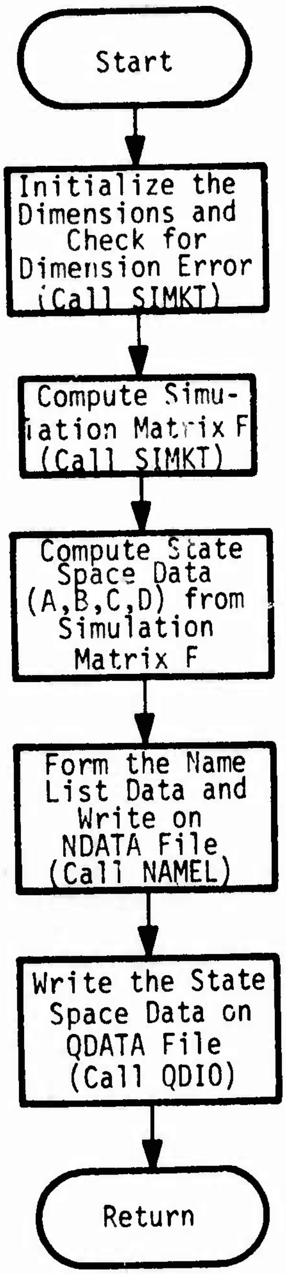

Figure 19. Subroutine STAMK2 Flow Chart

```
      SUBROUTINE STAMK2(V,W,F,XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1A,B,C,D,AT,BT,CT,DT,P,Q,R,S,PRINT,HS,NNS,VNS,DESS,UNITS,
     2NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,
     3MAXN,MAXM,NXM,NRM,NUM,NYM,MR,MTFB,MST,MT,MS1,MS2,MS3,MS4,NB)
C
C     PURPOSE - TO OBTAIN STATE MODEL FROM TRANSFER FN REPRESENTATION
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        DEPRMS
C        MPRS
C        ODIO
C        TDINVR
C        DEPRM
C        MPP
C        NAMEL
C        SIMKT
C
C     ARGUMENTS LIST
C        V                     V ARRAY FOR COMPUTING SIMULATION MATRIX
C        W                     W ARRAY FOR COMPUTING SIMULATION MATRIX
C        F                     SIMULATION MATRIX
C        XDOT                  ARRAY FOR STATE DERIVATIVES
C        X                     ARRAY FOR STATES
C        RI                    ARRAY FOR INTERNAL OUTPUTS
C        UI                    ARRAY FOR INTERNAL INPUTS
C        U                     ARRAY FOR EXTERNAL INPUTS
C        NNX                   ARRAY FOR STORING SYSTEM DIMENSION NX
C        NNR                   ARRAY FOR STORING SYSTEM DIMENSION NR
C        NNU                   ARRAY FOR STORING SYSTEM DIMENSION NU
C        A           IN/OUT    STATE TRANSITION MATRIX
C        B           IN/OUT    CONTROL INPUT MATRIX
C        C           IN/OUT    STATE OUTPUT MATRIX
C        D           IN/OUT    CONTROL OUTPUT MATRIX
C        AT                    ARRAY FOR STORING TR FUNCTION QUADRUPLE A
C        BT                    ARRAY FOR STORING TR FUNCTION QUADRUPLE B
C        CT                    ARRAY FOR STORING TR FUNCTION QUADRUPLE C
C        DT                    ARRAY FOR STORING TR FUNCTION QUADRUPLE D
C        P                     INTERCONNECTION QUADRUPLE
C        Q                     INTERCONNECTION QUADRUPLE
C        R                     INTERCONNECTION QUADRUPLE
C        S                     INTERCONNECTION QUADRUPLE
C        PRINT                 STORES TRANSFER FUNCTION DATA
C        HS                    TRANSFER FUNCTION DATA
C        NNS         IN/OUT    NUMBER ARRAY FOR STATE
C        VNS         IN/OUT    VARIABLE NAME ARRAY FOR STATE
C        DESS        IN/OUT    DESCRIPTION ARRAY FOR STATE
C        UNITS       IN/OUT    UNIT ARRAY FOR STATE
C        NNO         IN/OUT    NUMBER ARRAY FOR OUTPUT
C        VNO         IN/OUT    VARIABLE NAME ARRAY FOR OUTPUT
C        DESO        IN/OUT    DESCRIPTION ARRAY FOR OUTPUT
C        UNITO       IN/OUT    UNIT ARRAY FOR OUTPUT
C        NNI         IN/OUT    NUMBER ARRAY FOR INPUT
C        VNI         IN/OUT    VARIABLE NAME ARRAY FOR INPUT
C        DESI        IN/OUT    DESCRIPTION ARRAY FOR INPUT
C        UNITI       IN/OUT    UNIT ARRAY FOR INPUT
C        MAXN        INPUT     MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C        MAXM        INPUT     MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C        NXM         INPUT     MAXIMUM NUMBER OF STATES
C        NRM         INPUT     MAXIMUM NUMBER OF OUTPUTS
C        NUM         INPUT     MAXIMUM NUMBER OF INPUTS
```

Figure 20. Subroutine STAMK2 Program Listing

```
C         NYM        INPUT      MAXIMUM DIMENSION FOR INTERCONN EQUATIONS
C         MB         INPUT      MAXIMUM NO OF SUBSYSTEMS FOR COMBINING
C         MTFB       INPUT      MAX NO OF TRANSFER FN BLOCKS FOR COMBINING
C         MST        INPUT      MAX POWER OF S IN THE TRANSFER FUNCTION
C         MT         INPUT      MAX NO OF TERMS IN THE TRANSFER FUNCTION
C         MS1        INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C         MS2        INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C         MS3        INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C         MS4        INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C         NB         INPUT      MAXIMUM SYSTEM NO - IMPLICIT MODEL
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      COMMON /SC1/ S1(1)
C     DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB)
C     DIMENSION DESOO(NRM,10,MB),UNITOO(NRM,4,MB)
C     DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)
C     DIMENSION NXX(MB),NRR(MB),NUU(MB)
      DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
      DIMENSION XDOT(MST,MTFB),X(MST,MTFB),RI(1,MTFB),UI(1,MTFB)
      DIMENSION U(NUM),NNX(MTFB),NNR(MTFB),NNU(MTFB)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION AT(MST,MST,MTFB),BT(MST,1,MTFB)
      DIMENSION CT(1,MST,MTFB),DT(1,1,MTFB)
      DIMENSION P(MTFB,MTFB),Q(MTFB,NUM),R(NRM,MTFB),S(NRM,NUM)
      DIMENSION PRINT(2,MT),HS(2,MT,MTFB)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSTAM,4HK2  ,2,0,IW)
      L1=1 $ L2=L1+NXM*MB*10 $ L3=L2+NXM*MB*4 $ L4=L3+NRM*MB*10
      L5=L4+NRM*MB*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MB*4
      L8=L7+MB $ L9=L8+MB $ L10=L9+MB
      IF(L1 .GT.MS1)
     1CALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,2,0,4HSTAM,4HK2  ,IW)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HSTAM,4HK2  ,2,0,IW)
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0
      NXA=0 $ NRA=0 $ NUA=0
      EPSF=1.0E-30 $ T=0.0 $ INIT=0 $ NFLAG=0
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))CALL HPR(HEAD,IW)
C
C     INITIALIZING CALL TO SIMKT
C
      NX=0 $ NY=0 $ NR=0 $ NU=0
      N1=1 $ N2=N1+NX $ N3=N2+NY
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
     2MT,NUM,NRM,INIT,T)
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HSTAM,4HK2  ,2,0,IW)
C
C     CHECK FOR DIMENSION ERROR
C
      INIT = 1
      M=2*NX+NY+NU
      N=NX+NY+NR
      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))
     1CALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,2,0,4HSTAM,4HK2  ,IW)
      N1=1 $ N2=N1+NX $ N3=N2+NY
C
C     ZERO OUT XDOT,RI,UI,X,U
C
      DO 10 NN=1,NMAX
      MX=NNX(NN)
      DO 10 J=1,MX
      XDOT(J,NN)=0.0
```

Figure 20. Subroutine STAMK2 Program Listing (Continued)

```
10    X(J,NN)=0.0
      DO 11 NN=1,NMAX
      MX=NNR(NN)
      DO 12 J=1,MX
12    RI(J,NN)=0.0
      MX=NNU(NN)
      DO 13 J=1,MX
13    UI(J,NN)=0.0
11    CONTINUE
      DO 14 I=1,NU
14    U(I)=0.0
C
C     COMPUTE PARTIALS WRT STATE DERIVATIVES
C
      JJ=0
      DO 50 NN=1,NMAX
      MX=NNX(NN)
      DO 50 J=1,MX
      JJ=JJ+1
      XDOT(J,NN)=1.
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
     2MT,NUM,NRM,INIT,T)
      XDOT(J,NN)=0.
      DO 50 I=1,N
50    F(I,JJ)=V(I)
C
C     COMPUTE PARTIALS WRT INTERNAL OUTPUTS
C
      DO 100 NN=1,NMAX
      MX=NNR(NN)
      DO 100 J=1,MX
      JJ=JJ+1
      RI(J,NN)=1.
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
     2MT,NUM,NRM,INIT,T)
      RI(J,NN)=0.
      DO 100 I=1,N
100   F(I,JJ)=V(I)
C
C     COMPUTE PARTIALS WRT INTERNAL INPUTS
C
      DO 150 NN=1,NMAX
      MX=NNU(NN)
      DO 150 J=1,MX
      JJ=JJ+1
      UI(J,NN)=1.
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
     2MT,NUM,NRM,INIT,T)
      UI(J,NN)=0.
      DO 150 I=1,N
150   F(I,JJ)=V(I)
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HSTAM,4HK2  ,2,0,IW)
C
C     COMPUTE PARTIALS WRT STATES
C
      DO 201 NN=1,NMAX
      MX=NNX(NN)
      DO 201 J=1,MX
      JJ=JJ+1
      X(J,NN)=1.
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
     2MT,NUM,NRM,INIT,T)
```

Figure 20. Subroutine STAMK2 Program Listing (Continued)

```
      X(J,N+I)=0.
      DO 201 I=1,N
  201 F(I,JJ)=V(I)
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HSTAM,4HK2  ,2,0,IW)
C
C     COMPUTE PARTIALS WRT EXTERNAL INPUTS
C
      DO 251 J=1,NU
      JJ=JJ+1
      U(J)=1.
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTEB,MST,
     2MT,NUM,NRM,INIT,T)
      U(J)=0.
      DO 251 I=1,N
  251 F(I,JJ)=V(I)
 8002 CONTINUE
C
C     COMPUTE THE SIMULATION MATRIX
C
      NV=NX+NY
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIM )
      DO 51 I=1,NV
      DO 52 J=1,NV
   52 F(I,J)=-F(I,J)
   51 F(I,I)=F(I,I)+1.
      CALL TDINVR(ISOL,IDSOL,NV,-M,F,MAXN,KDUM,DET)
      IB=NV+1
      IE=NV+NR
      JB=IB
      JE=M
      DO 53 I=IB,IE
      DO 53 J=JB,JE
      DO 53 K=1,NV
   53 F(I,J)=F(I,J)+F(I,K)*F(K,J)
      DO 530 I=1,IE
      DO 530 J=1,JE
      IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0
  530 CONTINUE
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIMI)
C
C     FORM A,B,C,D MATRICES
C
      J1=NV+1
      J2=NV+NX
      J3=J1+NX
      J4=J2+NU
      I1=NV+1
      I2=NV+NR
      DO 6001 I=1,NX
      DO 6001 J=J1,J2
      JJ=J-J1+1
 6001 A(I,JJ)=F(I,J)
      DO 6002 I=1,NX
      DO 6002 J=J3,J4
      JJ=J-J3+1
 6002 B(I,JJ)=F(I,J)
      DO 6003 I=I1,I2
      II=I-I1+1
      DO 6003 J=J1,J2
      JJ=J-J1+1
 6003  C(II,JJ)=F(I,J)
      DO 6004 I=I1,I2
      II=I-I1+1
      DO 6004 J=J3,J4
      JJ=J-J3+1
```

Figure 20. Subroutine STAMK2 Program Listing (Continued)

```
 6004 D(II, IJ)=F(I,J)
C
C     READ AND UPDATE NAME LIST DATA
C
      KB=NMAX
      CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
     1DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
     2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)
      IF(IPRINT.EQ.6)CALL DEBUG(6,4HSTAM,4HK2  ,2,0,IW)
C
C     WRITE QUADRUPLE DATA ON FILE QDATA
C
      IQ=0
      MFLAG=2
      NXA=NX $ NRA=NR $ NUA=NU
      CALL QDIO(A,B,C,D,A,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,HEAD,MARK,
     2LOCATE,NULL,INSERT,MFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(7,4HSTAM,4HK2  ,2,0,IW)
      RETURN
      END
```

Figure 20. Subroutine STAMK2 Program Listing (Concluded)

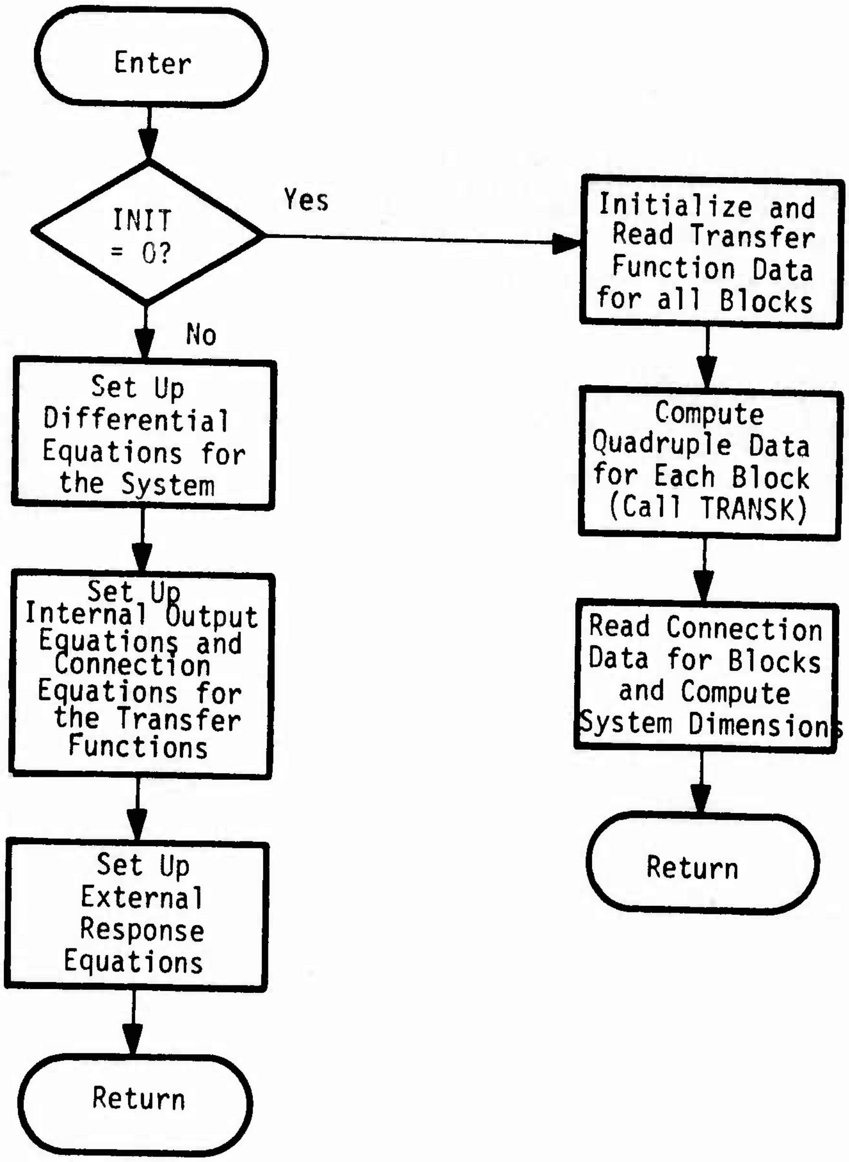

Figure 21. Subroutine SIMKT Flow Chart

```
      SUBROUTINE SIMKT(XDOTL,YL,RL,XDOT,X,RI,UI,U,NNX,NNR,NNU,
     1AT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
     2MT,NUM,NRM,INIT,T)
C
C     PURPOSE - TO COMBINE SUBSYSTEM BLOCKS WHICH ARE DESCRIBED BY
C     RATIONAL TRANSFER FUNCTIONS
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        INPT
C        PHERR
C        TRANSK
C        DFN
C        MPRS
C        TPR
C        ZERO
C
C     ARGUMENTS LIST
C        XDOTL       OUTPUT     ARRAY FOR DERIVATIVE OF STATE
C        YL          OUTPUT     ARRAY FOR Y EQUATION VARIABLES
C        RL          OUTPUT     ARRAY FOR EXTERNAL RESPONSE VARIABLES
C        NX          OUTPUT     NUMBER OF STATES
C        NY          OUTPUT     NUMBER OF Y EQUATIONS
C        NR          OUTPUT     NUMBER OF OUTPUTS
C        NU          OUTPUT     NUMBER OF INPUTS
C        NMAX                   BLOCK NO OF TRANSFER FUNCTION
C        INIT        INPUT      INITIAL MODE FLAG
C        T           OUTPUT     SAMPLE TIME
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      DIMENSION XDOTL(NX),YL(NY),RL(NR)
      DIMENSION XDOT(MST,MTFB),X(MST,MTFB),RI(1,MTFB),UI(1,MTFB)
      DIMENSION U(NUM),NNX(MTFB),NNR(MTFB),NNU(MTFB)
      DIMENSION AT(MST,MST,MTFB),BT(MST,1,MTFB)
      DIMENSION CT(1,MST,MTFB),DT(1,1,MTFB)
      DIMENSION P(MTFB,MTFB),Q(MTFB,NUM),R(NRM,MTFB),S(NRM,NUM)
      DIMENSION PRINT(2,MT),HS(2,MT,MTFB)
      DIMENSION CARD(20),IHEAD(3)
      REAL IHEAD
      DATA HC,HEND,HBLOC,HK,HDELA,HY/1HC,4HEND ,4HBLOC,1HK,4HDELA,1HY/
      DATA HUIR,HI,HUIU,HRRI,HRU/4HUI/R,4HI    ,4HUI/U,4HR/RI,4HR/U /
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSIMK,4HT    ,2,0,IW)
      IF(INIT.NE.0) GO TO 100
C
C     INITIALIZE
C
      NBK=0
      T=0.0
      DO 1020 I=1,2
      DO 1020 J=1,MT
      DO 1020 K=1,MTFB
 1020 HS(I,J,K)=0.0
      CALL ZERO(P,MTFB,MTFB)
      CALL ZERO(Q,MTFB,NUM)
      CALL ZERO(R,NRM,MTFB)
      CALL ZERO(S,NRM,NUM)
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,1030)
 1030 FORMAT(//,20X,41H*** TRANSFER FUNCTION DATA FOR BLOCKS ***,//)
 1040 CONTINUE
      READ(IR,1060)CARD
```

Figure 22. Subroutine SIMKT Program Listing

```
 1060 FORMAT(20A4)
      DECODE(4,1100,CARD(1))CC,DUMMY
 1100 FORMAT(A1,A3)
      IF(CC.EQ.HC)GO TO 1040
      IF(CARD(1).EQ.HEND)GO TO 1480
      IF(CARD(1).NE.HBLOC)GO TO 1440
      BLK1=CARD(1)
      DECODE(4,1120,CARD(2))BLK2,NBKNO,DUMMY
 1120 FORMAT(A1,I2,A1)
      BKD1=CARD(3)
      BKD2=CARD(4)
      IF(BLK2.NE.HK)GO TO 1440
      N=NBKNO
      NBK=NBK+1
      IF(NBK.GT.MTFB)GO TO 1190
      DO 1125 I=1,3
 1125 IHEAD(I)=CARD(I)
      IF((BKD1.EQ.HDELA).AND.(BKD2.EQ.HY))GO TO 1300
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HSIMK,4HT    ,2,0,IW)
C
C     READ RATIONAL TRANSFER FUNCTION DATA
C
      CALL ZERO(PRINT,2,MT)
      CALL INPT(PRINT,2,MT)
      DO 1130 I=1,2
      DO 1130 J=1,MT
 1130 HS(I,J,N)=PRINT(I,J)
      DO 1160 I=1,MT
      IF(HS(1,I,N).NE.0.0)NNX(N)=I-1
      IF(HS(2,I,N).NE.0.0)NNX(N)=I-1
 1160 CONTINUE
      IF(HS(2,1,N).NE.0.0)GO TO 1200
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,1180)
 1180 FORMAT(1H1,//,1X,37HTRANSFER FUNCTION SPECIFICATION ERROR)
      STOP 111
 1190 CONTINUE
      WRITE(IW,1195)
 1195 FORMAT(1H1,//,29HTOO MANY BLOCKS FOR COMBINING)
      STOP 111
C
C     PRINT THE TRANSFER FUNCTION
C
 1200 CONTINUE
      IF((IPRINT.NE.3).AND.(IPRINT.LT.5))GO TO 1040
      NN1=NNX(N)+1
      DO 1240 I=1,2
      DO 1240 J=1,NN1
 1240 PRINT(I,J)=HS(I,J,N)
      CALL TPR(PRINT,NN1,MT,IHEAD,T,IW)
      GO TO 1040
 1300 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HSIMK,4HT    ,2,0,IW)
C
C     READ TIME DELAY SPECIFICATION
C
      READ(IR,1320)TD,XX,XR,UU,OMEGM,DELPHM,ND,NN
 1320 FORMAT(6E12.6,2I2)
      IF(TD.GT.0.0)GO TO 1380
      IF(UU.EQ.0.0)GO TO 1340
      TD=(XX-XR)/UU
      IF(TD.LE.0.0)GO TO 1340
      GO TO 1380
 1340 CONTINUE
```

Figure 22. Subroutine SIMKT Program Listing (Continued)

```
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,1360)
 1360 FORMAT(1H1,//,1X,30HTIME DELAY SPECIFICATION ERROR)
      STOP 111
 1380 CONTINUE
      IF(OMEGM.NE.0.0)GO TO 1400
      IF((ND.EQ.0).OR.(NN.EQ.0))GO TO 1340
      CALL DFN(HS,MT,MTFB,ND,NN,N,TD,IPRINT,IW)
      NNX(N)=ND-1
      GO TO 1200
 1400 CONTINUE
      IF(DELPHM.LE.0.0)GO TO 1340
      DO 1420 ND=2,5
      NNX(N)=ND-1
      NDM=ND
      IF(ND.EQ.5)NDM=4
      DO 1420 NN=1,NDM
      CALL DFN(HS,MT,MTFB,ND,NN,N,TD,IPRINT,IW)
      CALL PHERR(HS,MT,MTFB,ND,N,OMEGM,TD,DELPH,IPRINT,IW)
      IF(DELPH.LE.DELPHM)GO TO 1200
 1420 CONTINUE
      WRITE(IW,1430)DELPHM,DELPH
 1430 FORMAT(1H1,//,1X,38HTIME DELAY SPECIFICATION CANNOT BE MET,//,
     11X,20HALLOWED PHASE ERROR=,E12.6,//,
     21X,20HACTUAL PHASE ERROR =,E12.6,//)
      GO TO 1200
 1440 CONTINUE
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,1460)
 1460 FORMAT(1H1,//,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)
      STOP 111
 1480 CONTINUE
      NMAX=NBK
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HSIMK,4HT   ,2,0,IW)
C
C     COMPUTE QUADRUPLES FOR ALL BLOCKS
C
      DO 1540 N=1,NMAX
      CALL TRANSK(NNX,NNR,NNU,AT,BT,CT,DT,PRINT,HS,
     1MST,MT,MTFB,N,NUM,NRM,IPRINT,IW)
 1540 CONTINUE
      NX=0
      DO 1560 N=1,NMAX
      NX=NX+NNX(N)
      NNU(N)=1
 1560 NNR(N)=1
      NY=2*NMAX
C
C     READ INTERCONNECTION QUADRUPLES AND PRINT THEM
C
 1580 CONTINUE
      READ(IR,1060)CARD
      DECODE(4,1100,CARD(1))CC,DUMMY
      IF(CC.EQ.HC)GO TO 1580
      IF(CARD(1).EQ.HEND)GO TO 1600
      IF((CARD(1).EQ.HUIR).AND.(CARD(2).EQ.HI))CALL INPT(P,MTFB,MTFB)
      IF((CARD(1).EQ.HUIR).AND.(CARD(2).EQ.HI))GO TO 1580
      IF(CARD(1).EQ.HUIU)CALL INPT(Q,MTFB,NUM)
      IF(CARD(1).EQ.HUIU)GO TO 1580
      IF(CARD(1).EQ.HRRI)CALL INPT(R,NRM,MTFB)
      IF(CARD(1).EQ.HRRI)GO TO 1580
      IF(CARD(1).EQ.HRU)CALL INPT(S,NRM,NUM)
```

Figure 22. Subroutine SIMKT Program Listing (Continued)

```
      IF(CARD(1).EQ.HRU)GO TO 1580
      GO TO 1440
 1600 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HSIMK,4HT   ,2,0,IW)
C
C CALCULATE NR AND NU
C
      DO 1660 J=1,NUM
      DO 1620 I=1,NMAX
      IF(Q(I,J).NE.0.0)GO TO 1660
 1620 CONTINUE
      DO 1640 I=1,NRM
      IF(S(I,J).NE.0.0)GO TO 1660
 1640 CONTINUE
      NU=J-1
      GO TO 1680
 1660 CONTINUE
      NU=NUM
 1680 CONTINUE
      IF(NU.EQ.0)GO TO 1780
      DO 1740 I=1,NRM
      DO 1700 J=1,NMAX
      IF(R(I,J).NE.0.0)GO TO 1740
 1700 CONTINUE
      DO 1720 J=1,NU
      IF(S(I,J).NE.0.0)GO TO 1740
 1720 CONTINUE
      NR=I-1
      GO TO 1760
 1740 CONTINUE
      NR=NRM
 1760 CONTINUE
      IF(NR.GT.0)GO TO 1820
 1780 CONTINUE
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,1800)
 1800 FORMAT(1H1,//,1X,35HINTERCONNECTION SPECIFICATION ERROR)
      STOP 111
 1820 CONTINUE
      IF((IPRINT.NE.3).AND.(IPRINT.LT.5))GO TO 1860
      WRITE(IW,1840)
 1840 FORMAT(//,20X,34H*** CONNECTION DATA FOR BLOCKS ***,//)
      CALL MPRS(P,MTFB,MTFB,NMAX,NMAX,T,4HP   )
      CALL MPRS(Q,MTFB,NUM,NMAX,NU,T,4HQ   )
      CALL MPRS(R,NRM,MTFB,NR,NMAX,T,4HR   )
      CALL MPRS(S,NRM,NUM,NR,NU,T,4HS   )
 1860 CONTINUE
      RETURN
  100 CONTINUE
C
C     COMPUTE SUBSYSTEM STATES  XDOT(N)=AN*XN +BN*UN
C
      II=0
      DO 251 N=1,NMAX
      MX=NNX(N)
      DO 200 I=1,MX
      II=II+1
      XDOTL(II)=0.0
      NUX = NNU(N)
      DO201 J=1,NUX
  201 XDOTL(II)=XDOTL(II)+BT(I,J,N)*UI(J,N)
      DO 200 J=1,MX
  200 XDOTL(II)=XDOTL(II)+AT(I,J,N)*X(J,N)
  251 CONTINUE
```

Figure 22. Subroutine SIMKT Program Listing (Continued)

```
C
C     COMPUTE INTERNAL OUTPUTS RIN=CN*AN+DN*UN
C
      II=0
      DO 350 N=1,NMAX
      MX=NNP(N)
      DO 300 I=1,MX
      II=II+1
      YL(II)=0.0
      MX1=NNX(N)
      DO 301 J=1,MX1
301   YL(II)=YL(II)+CT(I,J,N)*X(J,N)
      NX1=NNU(N)
      DO 300 J=1,NX1
300   YL(II)=YL(II)+DT(I,J,N)*UI(J,N)
350   CONTINUE
C
C     INTERCONNECTION EQUATIONS
C
      DO 240 I=1,NMAX
      II=II+1
      YL(II)=0.0
      DO 230 J=1,NMAX
  230 YL(II)=YL(II)+P(I,J)*RI(1,J)
      DO 240 J=1,NU
  240 YL(II)=YL(II)+Q(I,J)*U(J)
C
C     EXTERNAL RESPONS EQUATIONS
C
      II=0
      DO 280 I=1,NR
      II=II+1
      RL(II)=0.0
      DO 270 J=1,NMAX
  270 RL(II)=RL(II)+R(I,J)*RI(1,J)
      DO 280 J=1,NU
  280 RL(II)=RL(II)+S(I,J)*U(J)
      RETURN
      END
```

Figure 22. Subroutine SIMKT Program Listing (Concluded)

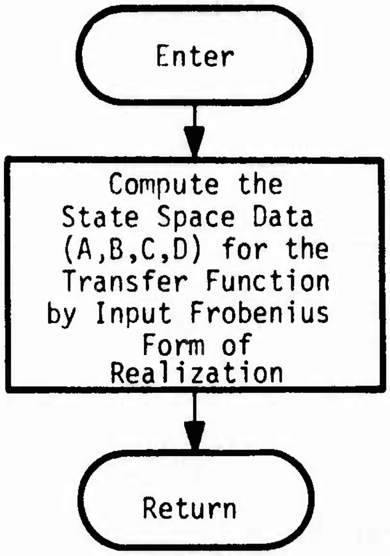

Figure 23. Subroutine TRANSK Flow Chart

```
      SUBROUTINE TRANSK(NNX,NNR,NNU,AT,BT,CT,DT,PRINT,HS,
     1MST,MT,MTFB,N,NUM,NRM,IPRINT,IW)
C
C     PURPOSE - TO COMPUTE QUADRUPLES FOR RATIONAL TRANSFER FUNCTION
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C
C     ARGUMENTS LIST
C        N          INPUT      TRANSFER FN BLOCK NO
C        IPRINT     INPUT      PRINT CONTROL FLAG
C        IW         INPUT      FILE NO FOR LINE PRINTER
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION NNX(MTFB),NNR(MTFB),NNU(MTFB)
      DIMENSION HS(2,MT,MTFB),PRINT(2,MT)
      DIMENSION AT(MST,MST,MTFB),BT(MST,1,MTFB)
      DIMENSION CT(1,MST,MTFB),DT(1,1,MTFB)
C
C     ZERO OUT STORAGE SPACE
C
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HTRAN,4HSK  ,2,0,IW)
      MX=NNX(N)
      MXM1=NNX(N)-1
      MXP1=NNX(N)+1
      DO 1 I=1,MX
      DO 1 J=1,MX
    1 AT(I,J,N)=0.0
      DO 2 I=1,MX
      J=1
    2 BT(I,J,N)=0.0
      DO 3 J=1,MX
      I=1
    3 CT(I,J,N)=0.0
      DT(1,1,N)=0.0
C
C     COMPUTE AT,BT
C
C     SET OFF DIAGONAL TERMS IN A TO UNITY
C
      DO 50 I=1,MXM1
   50 AT(I,I+1,N)=1.
      BT(NNX(N),1,N)=1./HS(2,1,N)
C
C     COMPUTE LAST ROW OF A
C
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HTRAN,4HSK  ,2,0,IW)
      DO 100 J=1,MX
  100 AT(NNX(N),J,N)=-HS(2,NNX(N)+2-J,N)*BT(NNX(N),1,N)
C
C     COMPUTE CT,DT
C
      DO 200 J=1,MX
  200 CT(1,J,N)=HS(1,NNX(N)+2-J,N)
      IF(HS(1,1,N).EQ.0.0) GO TO 400
      DO 300 J=1,MX
  300 CT(1,J,N)=CT(1,J,N)+AT(NNX(N),J,N)*HS(1,1,N)
      DT(1,1,N)=BT(NNX(N),1,N)*HS(1,1,N)
  400 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HTRAN,4HSK  ,2,0,IW)
      RETURN
      END
```

Figure 24. Subroutine TRANSK Program Listing

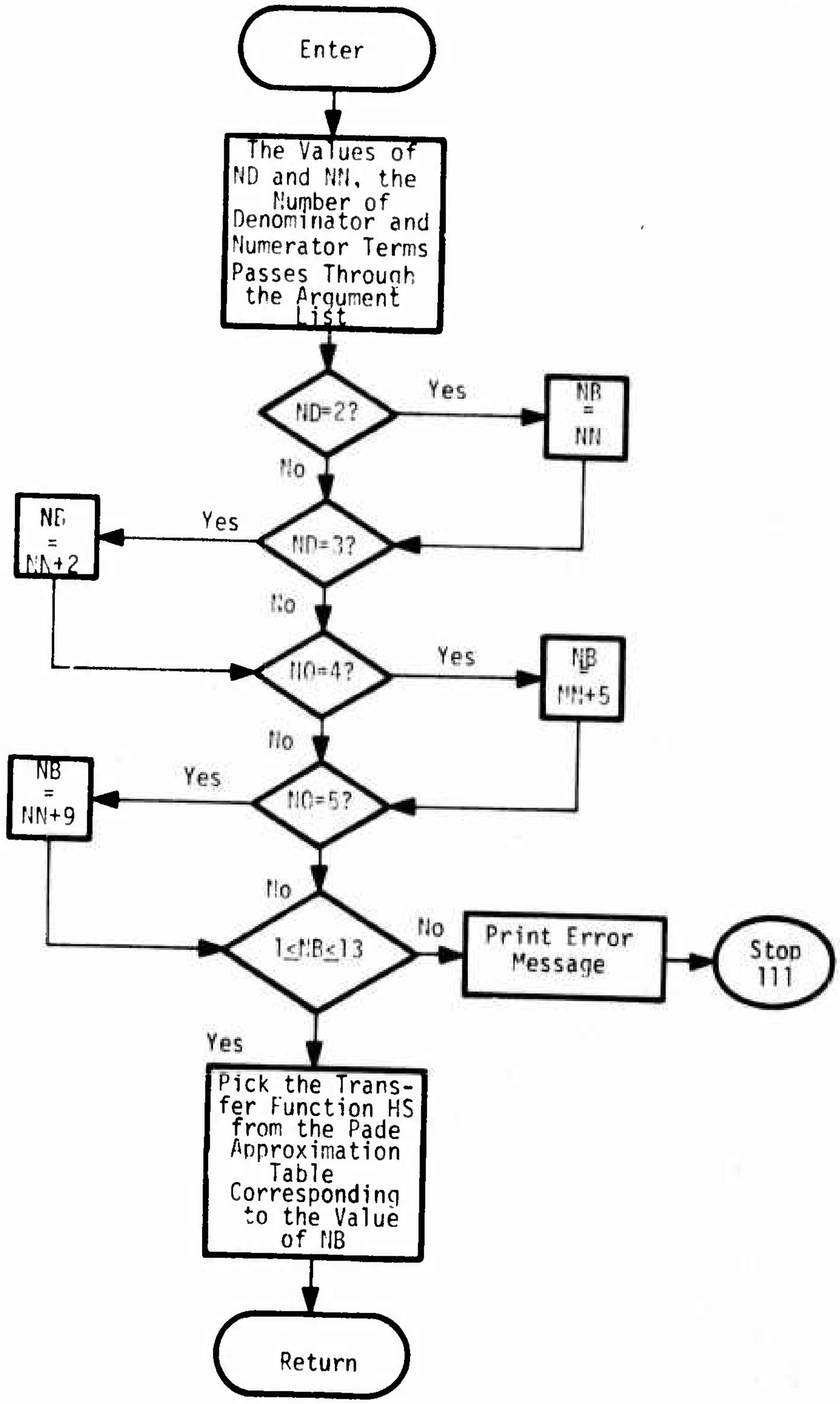

Figure 25. Subroutine DFN Flow Chart

```
      SUBROUTINE DFN(HS,MT,MTFB,ND,NN,N,TD,IPRINT,IW)
C
C     PURPOSE - TO PICK A PADE APPROXIMATION TO TIME DELAY
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        ND          INPUT       NO OF DENOMINATOR TERMS IN THE TR FN
C        NN          INPUT       NO OF NUMERATOR TERMS IN THE TR FN
C        N           INPUT       TRANSFER FN BLOCK NO
C        TD          INPUT       TIME OR TRANSPORT DELAY
C        IPRINT      INPUT       PRINT CONTROL FLAG
C        IW          INPUT       FILE NO FOR LINE PRINTER
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION HS(2,MT,MTFB)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HDFN ,4H    ,2,0,IW)
      IF(ND.EQ.2)NB=NN
      IF(ND.EQ.3)NB=NN+2
      IF(ND.EQ.4)NB=NN+5
      IF(ND.EQ.5)NB=NN+9
      IF((NB.GT.13).OR.(NB.LT.1))GO TO 660
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HDFN ,4H    ,2,0,IW)
      GO TO (510,520,530,540,550,560,570,580,590,600,610,620,630)NB
C
C     FIRST ORDER PADE APPROXIMATIONS
C
  510 CONTINUE
      HS(1,2,N)=1.0
      HS(2,1,N)=TD
      HS(2,2,N)=1.0
      GO TO 650
  520 CONTINUE
      HS(1,1,N)=-TD/2.0
      HS(1,2,N)=1.0
      HS(2,1,N)=TD/2.0
      HS(2,2,N)=1.0
      GO TO 650
C
C     SECOND ORDER PADE APPROXIMATIONS
C
  530 CONTINUE
      HS(1,3,N)=1.0
      HS(2,1,N)=TD*TD/2.0
      HS(2,2,N)=TD
      HS(2,3,N)=1.0
      GO TO 650
  540 CONTINUE
      HS(1,2,N)=-TD/3.0
      HS(1,3,N)=1.0
      HS(2,1,N)=TD*TD/6.0
      HS(2,2,N)=2.0*TD/3.0
      HS(2,3,N)=1.0
      GO TO 650
  550 CONTINUE
      HS(1,1,N)=TD*TD/12.0
      HS(1,2,N)=-TD/2.0
      HS(1,3,N)=1.0
      HS(2,1,N)=TD*TD/12.0
      HS(2,2,N)=TD/2.0
      HS(2,3,N)=1.0
      GO TO 650
C
```

Figure 26. Subroutine DFN Program Listing

```
C     THIRD ORDER PADE APPROXIMATIONS
C
  560 CONTINUE
      HS(1,4,N)=1.0
      HS(2,1,N)=TD**3/6.0
      HS(2,2,N)=TD*TD/2.0
      HS(2,3,N)=TD
      HS(2,4,N)=1.0
      GO TO 650
  570 CONTINUE
      HS(1,3,N)=-TD/4.0
      HS(2,1,N)=TD**3/24.0
      HS(1,4,N)=1.0
      HS(2,2,N)=TD*TD/4.0
      HS(2,3,N)=TD*3.0/4.0
      HS(2,4,N)=1.0
      GO TO 650
  580 CONTINUE
      HS(1,2,N)=TD*TD/20.0
      HS(1,3,N)=-2.0*TD/5.0
      HS(1,4,N)=1.0
      HS(2,1,N)=TD**3/60.0
      HS(2,2,N)=3.0*TD*TD/20.0
      HS(2,3,N)=TD*3.0/5.0
      HS(2,4,N)=1.0
      GO TO 650
  590 CONTINUE
      HS(1,1,N)=-TD**3/120.0
      HS(1,2,N)=TD*TD/10.0
      HS(1,3,N)=-TD/2.0
      HS(1,4,N)=1.0
      HS(2,1,N)=TD**3/120.0
      HS(2,2,N)=TD*TD/10.0
      HS(2,3,N)=TD/2.0
      HS(2,4,N)=1.0
      GO TO 650
C
C     FOURTH ORDER PADE APPROXIMATIONS
C
  600 CONTINUE
      HS(1,5,N)=1.0
      HS(2,1,N)=TD**4/24.0
      HS(2,2,N)=TD**3/6.0
      HS(2,3,N)=TD*TD/2.0
      HS(2,4,N)=TD
      HS(2,5,N)=1.0
      GO TO 650
  610 CONTINUE
      HS(1,4,N)=-TD/5.0
      HS(1,5,N)=1.0
      HS(2,1,N)=TD**4/120.0
      HS(2,2,N)=2.0*TD**3/30.0
      HS(2,3,N)=TD*TD*3.0/10.0
      HS(2,4,N)=TD*4.0/5.0
      HS(2,5,N)=1.0
      GO TO 650
  620 CONTINUE
      HS(1,3,N)=TD*TD/30.0
      HS(1,4,N)=-TD/3.0
      HS(1,5,N)=1.0
      HS(2,1,N)=TD**4/360.0
      HS(2,2,N)=TD**3/30.0
      HS(2,3,N)=TD*TD*2.0/10.0
      HS(2,4,N)=TD*2.0/3.0
      HS(2,5,N)=1.0
      GO TO 650
```

Figure 26. Subroutine DFN Program Listing (Continued)

```
  630 CONTINUE
      HS(1,2,N)=-TD**3/210.0
      HS(1,3,N)=TD*TD/14.0
      HS(1,4,N)=-3.0*TD/7.0
      HS(1,5,N)=1.0
      HS(2,1,N)=TD**4/840.0
      HS(2,2,N)=4.0*TD**3/210.0
      HS(2,3,N)=TD*TD*2.0/14.0
      HS(2,4,N)=TD*4.0/7.0
      HS(2,5,N)=1.0
  650 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HDFN ,4H    ,2,0,IW)
      RETURN
C
C     PRINT ERROR MESSEGE
C
  660 CONTINUE
      WRITE(IW,670)
  670 FORMAT(1H1,//,1X,43HDIMENSIONS FOR TIME DELAY EXCEEDS THE LIMIT)
      STOP 111
      END
```

Figure 26. Subroutine DFN Program Listing (Concluded)

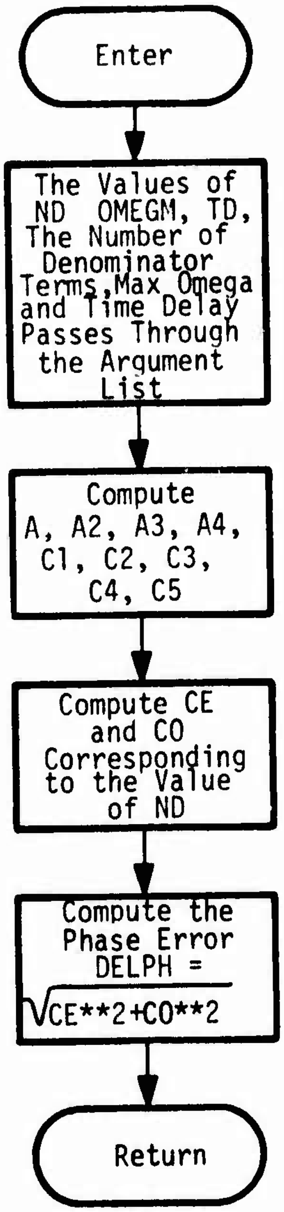

Figure 27. Subroutine PHERR Flow Chart

```
      SUBROUTINE PHERR(HS,MT,MTER,ND,N,OMEGM,TD,DELPH,IPRINT,IW)
C
C     PURPOSE - TO COMPUTE PHASE ERROR OF PADE APPROXIMATION TO TIME DEL
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C
C     ARGUMENTS LIST
C        ND          INPUT      NO OF DENOMINATOR TERMS IN THE TR FN
C        N           INPUT      TRANSFER FN BLOCK NO
C        OMEGM       INPUT      MAXIMUM FREQUENCY FOR COMPUTING PHASE ERROR
C        TD          INPUT      TIME OR TRANSPORT DELAY
C        DELPH       OUTPUT     PHASE ERROR
C        IPRINT      INPUT      PRINT CONTROL FLAG
C        IW          INPUT      FILE NO FOR LINE PRINTER
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION HS(2,MT,MTER)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HPHER,4HR    ,2,0,IW)
      A=OMEGM*TD
      A2=A**2
      A3=A**3
      A4=A**4
      C1=HS(2,1,N)*A-HS(1,1,N)
      C2=HS(2,2,N)*A-HS(1,2,N)
      C3=HS(2,3,N)*A-HS(1,3,N)
      C4=HS(2,4,N)*A-HS(1,4,N)
      C5=HS(2,5,N)*A-HS(1,5,N)
      NO=ND-1
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HPHER,4HR    ,2,0,IW)
      GO TO(110,120,130,140)NO
  110 CONTINUE
      CE=C2
      CO=C1*A
      GO TO 150
  120 CONTINUE
      CE=C3-C1*A2
      CO=C2*A
      GO TO 150
  130 CONTINUE
      CE=C4-C2*A2
      CO=C3*A-C1*A3
      GO TO 150
  140 CONTINUE
      CE=C5-C3*A2+C1*A4
      CO=C4*A-C2*A3
  150 CONTINUE
      CE2=CE**2
      CO2=CO**2
      DELPH=SQRT(CE2+CO2)
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HPHER,4HR    ,2,0,IW)
      RETURN
      END
```

Figure 28. Subroutine PHERR Program Listing

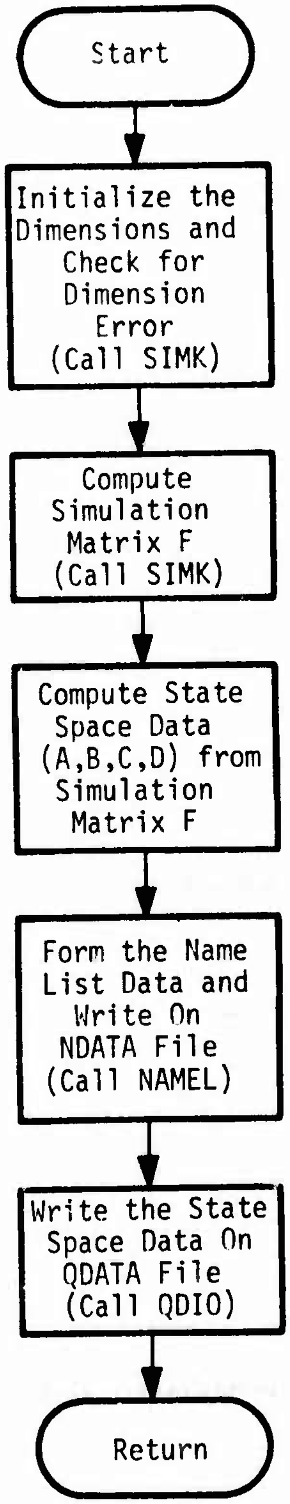

Figure 29. Subroutine STAMK3 Flow Chart

```
      SUBROUTINE STAMK3(V,W,F,XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1A,B,C,D,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     2NNI,VNI,DESI,UNITI,MAXN,MAXM,NXM,NRM,NUM,NYM,
     3MB,MN,MM,MP,MQ,MR,MS1,MS2,MS3,MS4,NB,NRMMB)
C
C     PURPOSE - TO OBTAIN STATE SPACE MODEL FROM INTERCONNECTION
C     DATA FOR SUBSYSTEMS OR TO READ DIRECTLY THE STATE SPACE DATA
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEPRM
C        MPR
C        NAMEL
C        QUADK
C        TDINVR
C        DEPRMS
C        MPRS
C        QDIO
C        SIMK
C
C     ARGUMENTS LIST
C        V                     V ARRAY FOR COMPUTING SIMULATION MATRIX
C        W                     W ARRAY FOR COMPUTING SIMULATION MATRIX
C        F                     SIMULATION MATRIX
C        XDOT                  ARRAY FOR STATE DERIVATIVES
C        X                     ARRAY FOR STATES
C        RI                    ARRAY FOR INTERNAL OUTPUTS
C        UI                    ARRAY FOR INTERNAL INPUTS
C        U                     ARRAY FOR EXTERNAL INPUTS
C        RIN                   ARRAY FOR INTERNAL OUTPUTS FOR ALL SYSTEMS
C        NNX                   ARRAY FOR STORING SYSTEM DIMENSION NX
C        NNR                   ARRAY FOR STORING SYSTEM DIMENSION NR
C        NNU                   ARRAY FOR STORING SYSTEM DIMENSION NU
C        A          IN/OUT     STATE TRANSITION MATRIX
C        B          IN/OUT     CONTROL INPUT MATRIX
C        C          IN/OUT     STATE OUTPUT MATRIX
C        D          IN/OUT     CONTROL OUTPUT MATRIX
C        NNS        IN/OUT     NUMBER ARRAY FOR STATE
C        VNS        IN/OUT     VARIABLE NAME ARRAY FOR STATE
C        DESS       IN/OUT     DESCRIPTION ARRAY FOR STATE
C        UNITS      IN/OUT     UNIT ARRAY FOR STATE
C        NNO        IN/OUT     NUMBER ARRAY FOR OUTPUT
C        VNO        IN/OUT     VARIABLE NAME ARRAY FOR OUTPUT
C        DESO       IN/OUT     DESCRIPTION ARRAY FOR OUTPUT
C        UNITO      IN/OUT     UNIT ARRAY FOR OUTPUT
C        NNI        IN/OUT     NUMBER ARRAY FOR INPUT
C        VNI        IN/OUT     VARIABLE NAME ARRAY FOR INPUT
C        DESI       IN/OUT     DESCRIPTION ARRAY FOR INPUT
C        UNITI      IN/OUT     UNIT ARRAY FOR INPUT
C        MAXN       INPUT      MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C        MAXM       INPUT      MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C        NXM        INPUT      MAXIMUM NUMBER OF STATES
C        NRM        INPUT      MAXIMUM NUMBER OF OUTPUTS
C        NUM        INPUT      MAXIMUM NUMBER OF INPUTS
C        NYM        INPUT      MAXIMUM DIMENSION FOR INTERCONN EQUATIONS
C        MB         INPUT      MAXIMUM NO OF SUBSYSTEMS FOR COMBINING
C        MN         INPUT      MM*MB
C        MM         INPUT      MAXIMUM OF (NRM,NUM)
C        MP         INPUT      MAXIMUM DIMENSION FOR P ARRAY
C        MQ         INPUT      MAXIMUM DIMENSION FOR Q ARRAY
C        MR         INPUT      MAXIMUM DIMENSION FOR R ARRAY
C        MS1        INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
```

Figure 30. Subroutine STAMK3 Program Listing

```
C          MS2         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C          MS3         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C          MS4         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C          NB          INPUT      MAXIMUM SYSTEM NO - IMPLICIT MODEL
C          NRMMB       INPUT MAXIMUM DIMENSION FOR RIN
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      COMMON /SC1/ S1(1)
C     DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB)
C     DIMENSION DESOO(NRM,10,MB),UNITOO(NRM,4,MB)
C     DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)
C     DIMENSION NXX(MB),NRR(MB),NUU(MB)
      COMMON /SC2/ S2(1)
C     DIMENSION ATC(NXM,NXM,MB),BTC(NXM,NUM,MB)
C     DIMENSION CTC(NRM,NXM,MB),DTC(NRM,NUM,MB)
C     DIMENSION PC(MN,MN),QC(MN,NUM),RC(NRM,MN),SC(NRM,NUM)
      COMMON /SC3/ S3(1)
C     DIMENSION PP(MP,MM,MM),QQ(MQ,MM,NUM),RR(MR,NRM,MM)
C     DIMENSION NSP(MP),NSQ(MQ),NSR(MR)
      DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
      DIMENSION XDOT(NXM,MB),X(NXM,MB),RI(NRM,MB),UI(NUM,MB)
      DIMENSION RIN(NRMMB),U(NUM),NNX(MB),NNR(MB),NNU(MB)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DATA HDINT/4HSINT/
C
C     PRINT SYSTEM DIMENSIONS IF NEEDED
C
      IF(IPRINT.EQ.6)WRITE(IW,165)MS1,MS2,MS3,MS4,MAXN,MAXM
     1,NXM,NRM,NUM,NYM,MM,MP,MQ,MR,MB,NB,MS,MN
  165 FORMAT(1X,15(I5,1X))
C
C     COMPUTE ARRAY START INDEXES
C
C
C     FOR DESSS,UNITSS,DESOO,UNITOO,DESII,UNITII,NXX,NRR,NUU
C
      L1=1 $ L2=L1+NXM*MB*10 $ L3=L2+NXM*MB*4 $ L4=L3+NRM*MB*10
      L5=L4+NRM*MB*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MB*4
      L8=L7+MB $ L9=L8+MB $ L10=L9+MB
C
C     FOR ATC,BTC,CTC,DTC,PC,QC,RC,SC
C
      M1=1 $ M2=M1+NXM*NXM*MB $ M3=M2+NXM*NUM*MB $ M4=M3+NRM*NXM*MB
      M5=M4+NRM*NUM*MB $ M6=M5+MN*MN $ M7=M6+MN*NUM $ M8=M7+NRM*MN
      M9=M8+NRM*NUM
C
C     FOR PP,QQ,RR,NSP,NSQ,NSR
C
      K1=1 $ K2=K1+MP*MM*MM $ K3=K2+MQ*MM*NUM $ K4=K3+MR*NRM*MM
      K5=K4+MP $ K6=K5+MQ $ K7=K6+MR
      IF(IPRINT.EQ.6)WRITE(IW,165)L1,L2,L3,L4,L5,L6,L7,L8,L9,L10
      IF(IPRINT.EQ.6)WRITE(IW,165)M1,M2,M3,M4,M5,M6,M7,M8,M9
      IF(IPRINT.EQ.6)WRITE(IW,165)K1,K2,K3,K4,K5,K6,K7
C
C     CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
C
      IF((L10.GT.MS1).OR.(M9.GT.MS2).OR.(K7.GT.MS3))
     1CALL DERRM(L10,M9,K7,MS4,MS1,MS2,MS3,MS4,3,0,4HSTAM,4HK3  ,IW)
      IF(SCODE.EQ.HDINT)GO TO 5
      CALL QUADK(A,B,C,D,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     1NNI,VNI,DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
```

Figure 30. Subroutine STAMK3 Program Listing (Continued)

```
     2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,MB,NB)
      RETURN
    5 CONTINUE
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0
      NXA=0 $ NRA=0 $ NUA=0
      EPSF=1.0E-30 $ T=0.0 $ INIT=0 $ NFLAG=1
      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))CALL HPR(HEAD,IW)
C
C     INITIALIZING CALL TO SUBROUTINE SIMK
C
      NX=0 $ NR=0 $ NU=0 $ NY=0
      N1=1 $ N2=N1+NX $ N3=N2+NY
      CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
     2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
     3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
C
C     CHECK FOR DIMENSION ERROR
C
      INIT = 1
      M=2*NX+NY+NU
      N=NX+NY+NR
      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))
     1CALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,3,0,4HSTAM,4HK3  ,IW)
      N1=1 $ N2=N1+NX $ N3=N2+NY
C
C     ZERO OUT XDOT,RI,UI,X,U
C
      DO 10 NN=1,NMAX
      MX=NNX(NN)
      DO 10 J=1,MX
      XDOT(J,NN)=0.0
   10 X(J,NN)=0.0
      DO 11 NN=1,NMAX
      MX=NNR(NN)
      DO 12 J=1,MX
   12 RI(J,NN)=0.0
      MX=NNU(NN)
      DO 13 J=1,MX
   13 UI(J,NN)=0.0
   11 CONTINUE
      DO 14 I=1,NU
   14 U(I)=0.0
C
C     COMPUTE PARTIALS WRT STATE DERIVATIVES
C
   45 CONTINUE
      JJ=0
      DO 50 NN=1,NMAX
      MX=NNX(NN)
      DO 50 J=1,MX
      JJ=JJ+1
      XDOT(J,NN)=1.
      CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
     2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
     3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
      XDOT(J,NN)=0.
      DO 50 I=1,N
   50 F(I,JJ)=V(I)
C
C     COMPUTE PARTIALS WRT INTERNAL OUTPUTS
C
      DO 100 NN=1,NMAX
      MX=NNR(NN)
      DO 100 J=1,MX
```

Figure 30. Subroutine STAMK3 Program Listing (Continued)

```
      JJ=JJ+1
      RI(J,NN)=1.
      CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
     2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
     3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
      RI(J,NN)=0.
      DO 100 I=1,N
100   F(I,JJ)=V(I)
C
C     COMPUTE PARTIALS WRT INTERNAL INPUTS
C
      DO 15 NN=1,NMAX
      MX=NNU(NN)
      DO 15 J=1,MX
      JJ=JJ+1
      UI(J,NN)=1.
      CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
     2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
     3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
      UI(J,NN)=0.
      DO 15 I=1,N
150   F(I,JJ)=V(I)
C
C     COMPUTE PARTIALS WRT STATES
C
      DO 201 NN=1,NMAX
      MX=NNX(NN)
      DO 201 J=1,MX
      JJ=JJ+1
      X(J,NN)=1.
      CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
     2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
     3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
      X(J,NN)=0.
      DO 201 I=1,N
  201 F(I,JJ)=V(I)
C
C     COMPUTE PARTIALS WRT EXTERNAL INPUTS
C
      DO 251 J=1,NU
      JJ=JJ+1
      U(J)=1.
      CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
     2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
     3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
      U(J)= .
      DO 251 I=1,N
  251 F(I,JJ)=V(I)
 8002 CONTINUE
C
C     COMPUTE THE SIMULATION MATRIX
C
      NV=NX+NY
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSTM )
      DO 51 I=1,NV
      DO 52 J=1,NV
   52 F(I,J)=-F(I,J)
   51 F(I,I)=F(I,I)+1.
C
C     XDOT ARRAY IS BEING USED AS A SCRATCH ARRAY IN TDINVR
C
      CALL TDINVR(ISOL,IDSOL,NV,-M,F,MAXN,XDOT,DET)
```

Figure 30. Subroutine STAMK3 Program Listing (Continued)

```
      IB=NV+1
      IE=NV+NR
      JB=IB
      JE=M
      DO 53 I=IB,IE
      DO 53 J=JB,JE
      DO 53 K=1,NV
   53 F(I,J)=F(I,J)+F(I,K)*F(K,J)
      DO 530  I=1,IE
      DO 530  J=1,JE
      IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0
  530 CONTINUE
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIMI)
C
C     FORM A,B,C,D MATRICES
C
      J1=NV+1
      J2=NV+NX
      J3=J1+NX
      J4=J2+NU
      I1=NV+1
      I2=NV+NR
      DO 6001 I=1,NX
      DO 6001 J=J1,J2
      JJ=J-J1+1
 6001 A(I,JJ)=F(I,J)
      DO 6002 I=1,NX
      DO 6002 J=J3,J4
      JJ=J-J3+1
 6002 B(I,JJ)=F(I,J)
      DO 6003 I=I1,I2
      II=I-I1+1
      DO 6003 J=J1,J2
      JJ=J-J1+1
 6003  C(II,JJ)=F(I,J)
      DO 6004 I=I1,I2
      II=I-I1+1
      DO 6004 J=J3,J4
      JJ=J-J3+1
 6004 D(II,JJ)=F(I,J)
C
C     UPDATE NAME LIST DATA
C
      KB=NMAX
      CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
     1DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
     2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)
C
C     WRITE QUADRUPLE DATA ON FILE QDATA
C
      IQ=0
      MFLAG=2
      CALL QDIO(A,B,C,D,S1,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JU,HEAD,MARK,
     2LOCATE,NULL,INSERT,MFLAG)
      RETURN
      END
```

Figure 30. Subroutine STAMK3 Program Listing (Concluded)

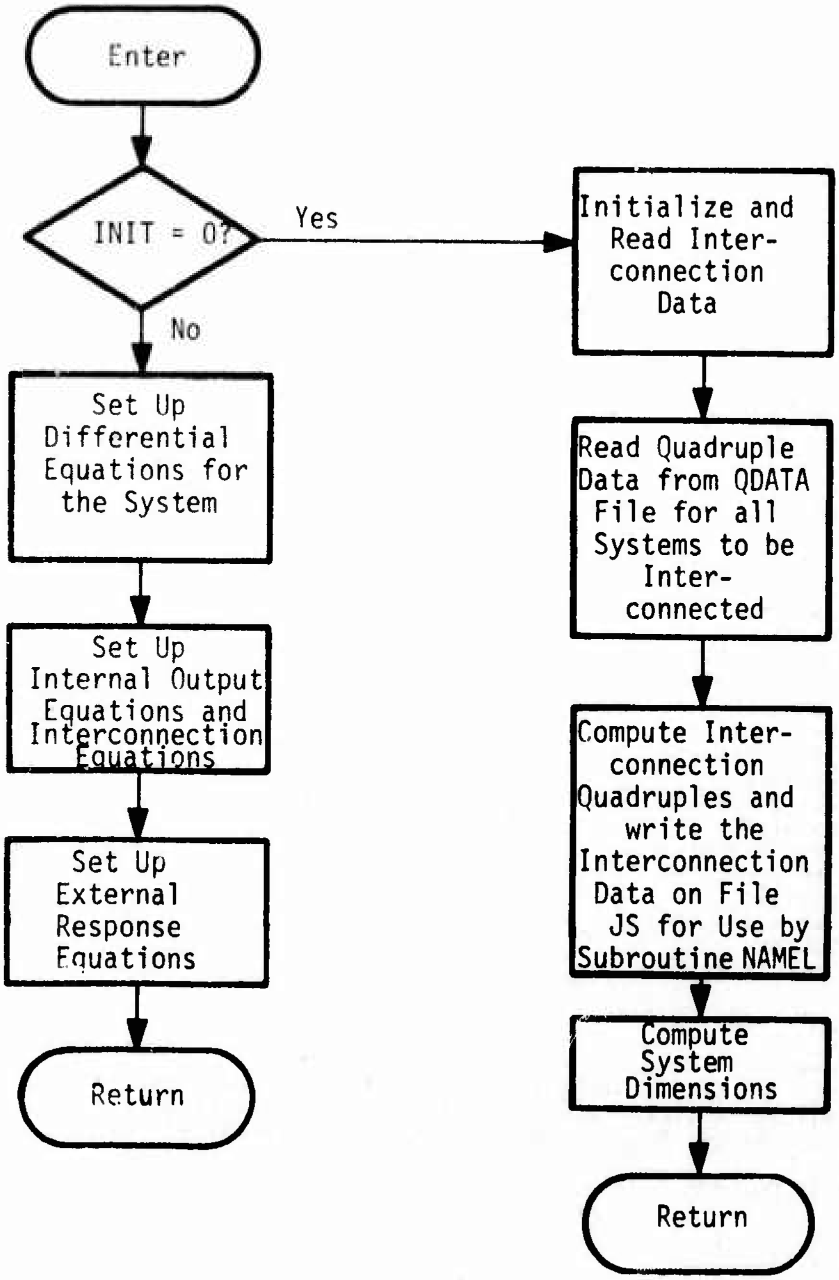

Figure 31. Subroutine SIMK Flow Chart

```
      SUBROUTINE SIMK(XDOTL,YL,RL,XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
     1AT,BT,CT,DT,P,Q,R,S,PP,QQ,RR,NSP,NSQ,NSR,NX,NY,NR,NU,KB,MB,
     2MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
C
C     PURPOSE - TO COMBINE SYSTEMS DESCRIBED BY QUADRUPLES
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        FILE
C        MPRS
C        INPT
C        ZERO
C
C     ARGUMENTS LIST
C        XDOTL       OUTPUT    ARRAY FOR DERIVATIVE OF STATE
C        YL          OUTPUT    ARRAY FOR Y EQUATION VARIABLES
C        RL          OUTPUT    ARRAY FOR EXTERNAL RESPONSE VARIABLES
C        AT                    ARRAY FOR STORING SUBSYSTEM QUADRUPLE A
C        BT                    ARRAY FOR STORING SUBSYSTEM QUADRUPLE B
C        CT                    ARRAY FOR STORING SUBSYSTEM QUADRUPLE C
C        DT                    ARRAY FOR STORING SUBSYSTEM QUADRUPLE D
C        P                     INTERCONNECTION QUADRUPLE
C        Q                     INTERCONNECTION QUADRUPLE
C        R                     INTERCONNECTION QUADRUPLE
C        S                     INTERCONNECTION QUADRUPLE
C        PP                    ARRAY FOR INTERMEDIATE INTERCONN DATA
C        QQ                    ARRAY FOR INTERMEDIATE INTERCONN DATA
C        RR                    ARRAY FOR INTERMEDIATE INTERCONN DATA
C        NSP                   ARRAY FOR INTERMEDIATE INTERCONN DATA
C        NSQ                   ARRAY FOR INTERMEDIATE INTERCONN DATA
C        NSR                   ARRAY FOR INTERMEDIATE INTERCONN DATA
C        NX          OUTPUT    NUMBER OF STATES
C        NY          OUTPUT    NUMBER OF Y EQUATIONS
C        NR          OUTPUT    NUMBER OF OUTPUTS
C        NU          OUTPUT    NUMBER OF INPUTS
C        INIT        INPUT     INITIAL MODE FLAG
C        T           OUTPUT    SAMPLE TIME
C        NXA         OUTPUT    NO OF STATES WITHOUT THE IMPLICIT MODEL
C        NRA         OUTPUT    NO OF OUTPUTS WITHOUT THE IMPLICIT MODEL
C        NUA         OUTPUT    NO OF INPUTS WITHOUT THE IMPLICIT MODEL
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(26)
      DIMENSION XDOTL(NX),YL(NY),RL(NR)
      DIMENSION XDOT(NXM,MB),X(NXM,MB),RI(NRM,MB),UI(NUM,MB)
      DIMENSION RIN(NRMMB),U(NUM),NNX(MB),NNR(MB),NNU(MB)
      DIMENSION AT(NXM,NXM,MB),BT(NXM,NUM,MB)
      DIMENSION CT(NRM,NXM,MB),DT(NRM,NUM,MB)
      DIMENSION P(MN,MN),Q(MN,NUM),R(NRM,MN),S(NRM,NUM)
      DIMENSION PP(MP,MM,MM),QQ(MQ,MM,NUM),RR(MR,NRM,MM)
      DIMENSION NSP(MP),NSQ(MQ),NSR(MR)
      DIMENSION CARD(20)
      DATA HC,HEND,HRI,HRS,HUI,HU/1HC,4HEND ,2HRI,2HR/,2HUI,1HU/
      DATA HOUTP,HINPU/4HOUTP,4HINPU/
      IF(INIT.NE.0) GO TO 100
C
C     INITIALIZE
C
      KP=0 $ KQ=0 $ KR=0
      KB=0
```

Figure 32. Subroutine SIMK Program Listing

```
      T=0.0
      NX1=0 $ NR1=0 $ NU1=0
      IF(IPRINT.EQ.6)WRITE(IW,900)KP,MB,MM,MP,MQ,MR,MS,NUM,NRM,MN,NB
  900 FORMAT(15(1X,I2))
      DO 960 I=1,NB
  960 NSYS(I)=-1
      DO 970 I=1,MP
  970 NSP(I)=0
      DO 980 I=1,MQ
  980 NSQ(I)=0
      DO 990 I=1,MR
  990 NSR(I)=0
      CALL ZERO(P,MN,MN)
      CALL ZERO(Q,MN,NUM)
      CALL ZERO(R,NRM,MN)
      CALL ZERO(S,NRM,NUM)
      DO 1010 J=1,MM
      DO 1000 K=1,MM
      DO 1000 I=1,MP
 1000 PP(I,J,K)=P(J,K)
      DO 1010 K=1,NUM
      DO 1010 I=1,MQ
 1010 QQ(I,J,K)=Q(J,K)
      DO 1020 J=1,NRM
      DO 1020 K=1,MM
      DO 1020 I=1,MR
 1020 RR(I,J,K)=R(J,K)
C
C     READ INTERCONNECTION DATA
C
 1040 CONTINUE
      READ(IR,1060)CARD
 1060 FORMAT(20A4)
      DECODE(4,1100,CARD(1))CC,DUMMY
 1100 FORMAT(A1,A3)
 1120 FORMAT(A2,I1,A1)
 1140 FORMAT(A2,A2)
      IF(CC.EQ.HC)GO TO 1040
      IF(CARD(1).EQ.HEND)GO TO 1480
      DECODE(4,1140,CARD(1))CODE1,CODE2
      IF(CODE1.EQ.HRS)GO TO 1200
      DECODE(4,1120,CARD(1))CODE1,NSY1,DUMMY
      IF(CODE1.EQ.HUI)GO TO 1160
      GO TO 1400
C
C     UPDATE THE SUSTEM NUMBER COUNTER NSYS
C
 1160 CONTINUE
      IF(KB.EQ.0)GO TO 1175
      DO 1170 N=1,KB
      IF(NSY1.EQ.NSYS(N))GO TO 1180
 1170 CONTINUE
 1175 CONTINUE
      KB=KB+1
      IF(KB.GT.MB)GO TO 1440
      NSYS(KB)=NSY1
 1180 CONTINUE
      DECODE(4,1120,CARD(2))CODE2,NSY2,DUMMY
      IF(CODE2.EQ.HRI)GO TO 1320
      DECODE(4,1100,CARD(2))CODE2,DUMMY
      IF(CODE2.EQ.HU)GO TO 1360
      GO TO 1400
 1200 CONTINUE
      IF(CODE2.EQ.HRI)GO TO 1220
      IF(CODE2.EQ.HU)GO TO 1280
      GO TO 1400
```

Figure 32. Subroutine SIMK Program Listing (Continued)

```
C
C     READ R MATRIX (R/RIM) INTO PROPER AREA OF RR MATRIX
C
 1220 CONTINUE
      DECODE(4,1240,CARD(2))NSY2,DUMMY
 1240 FORMAT(I1,A3)
      KR=KR+1
      IF(KR.GT.MR)GO TO 1470
      NSR(KR)=NSY2
      CALL ZERO(R,NRM,MN)
      CALL INPT(R,NRM,MN)
      DO 1260 I=1,NRM
      DO 1260 J=1,MM
 1260 RR(KR,I,J)=R(I,J)
      GO TO 1040
C
C     READ S MATRIX (R/U)
C
 1280 CONTINUE
      CALL ZERO(S,NRM,NUM)
      CALL INPT(S,NRM,NUM)
      GO TO 1040
C
C     READ P MATRIX (UIN/RIM) INTO PROPER AREA OF PP MATRIX
C
 1320 CONTINUE
      NSY=NQ*(NSY1-1)+NSY2
      KP=KP+1
      IF(KP.GT.MP)GO TO 1470
      NSP(KP)=NSY
      CALL ZERO(P,MN,MN)
      CALL INPT(P,MN,MN)
      DO 1340 I=1,MM
      DO 1340 J=1,MM
 1340 PP(KP,I,J)=P(I,J)
      IF(IPRINT.LT.6)GO TO 1040
      WRITE(IW,900)KP,NSY,NSP
      CALL MPRS(P,MN,MN,MM,MM,0,0,4HPP  )
      GO TO 1040
C
C     READ Q MATRIX (UIN/U) INTO PROPER AREA OF QQ MATRIX
C
 1360 CONTINUE
      KQ=KQ+1
      IF(KQ.GT.MQ)GO TO 1470
      NSQ(KQ)=NSY1
      CALL ZERO(Q,MN,NUM)
      CALL INPT(Q,MN,NUM)
      DO 1380 I=1,MM
      DO 1380 J=1,NUM
 1380 QQ(KQ,I,J)=Q(I,J)
      GO TO 1040
 1400 CONTINUE
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,1420)
 1420 FORMAT(1H1,//,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)
      STOP 111
 1440 CONTINUE
      WRITE(IW,1460)KR,MR
 1460 FORMAT(1H1,//,1X,30HTOO MANY SYSTEMS FOR COMBINING,
     1//,1X,5HKB = ,I2,5X,5HMR = ,I2)
      STOP 111
 1470 CONTINUE
      WRITE(IW,1475)
```

Figure 32. Subroutine SIMK Program Listing (Continued)

```
 1475 FORMAT(1H1,//,1X,39HTOO MANY INTERCONNECTIONS FOR COMBINING)
      STOP 111
C
C     OBTAIN QUADRUPLE DATA FOR SUBSYSTEMS FROM Q DATA FILE
C
 1480 CONTINUE
      DO 1490 I=1,20
 1490 CARD(I)=HEAD(I)
      DO 1520 N=1,KB
      NSY=NSYS(N)
      DO 1500 I=1,20
      HEAD(I)=SHEAD(NSY,I)
 1500 CONTINUE
      CALL FILE(JQ,LOCATE,HEAD)
      READ(JQ)T,NNXN,NNRN,NNUN,
     1        ((AT(I,J,N),I=1,NNXN),J=1,NNXN),
     2        ((BT(I,J,N),I=1,NNXN),J=1,NNUN),
     3        ((CT(I,J,N),I=1,NNRN),J=1,NNXN),
     4        ((DT(I,J,N),I=1,NNRN),J=1,NNUN)
      NNX(N)=NNXN
      NNR(N)=NNRN
      NNU(N)=NNUN
C
C     STORE THE IMPLICIT MODEL SYSTEM DIMENSIONS SEPARATELY
C
      IF(NSY.NE.NB)GO TO 1510
      NX1=NNXN
      NR1=NNRN
      NU1=NNUN
 1510 CONTINUE
 1520 CONTINUE
      DO 1530 I=1,20
 1530 HEAD(I)=CARD(I)
C
C     FORM INTERCONNECTION QUADRUPLES
C
      CALL ZERO(P,MN,MN)
      CALL ZERO(Q,MN,NUM)
      CALL ZERO(R,NRM,MN)
C
C     FORM R MATRIX (R/RI)
C
      KYOUT=0
      NM1=1
      NM2=0
      DO 1565 M=1,KB
      KYOUT=KYOUT+NNR(M)
      IF(M.GT.1)NM1=NM1+NNR(M-1)
      NM2=NM2+NNR(M)
      NSY2=NSYS(M)
      DO 1533 KR=1,MR
      IF(NSR(KR).EQ.NSY2)GO TO 1536
 1533 CONTINUE
      GO TO 1545
 1536 CONTINUE
      DO 1540 I=1,NRM
      DO 1540 J=NM1,NM2
      JJ=J-NM1+1
 1540 R(I,J)=RR(KR,I,JJ)
 1545 CONTINUE
C
C     FORM P MATRIX (UI/RI)
C
      NN1=1
      NN2=0
      DO 1562 N=1,KB
```

Figure 32. Subroutine SIMK Program Listing (Continued)

```
      IF(N.GT.1)NN1=NN1+NNU(N-1)
      NN2=NN2+NNU(N)
      NSY1=NSYS(N)
      NSY=NR*(NSY1-1)+NSY2
      DO 1550 KP=1,MP
      IF(NSP(KP).EQ.NSY)GO TO 1555
 1550 CONTINUE
      GO TO 1562
 1555 CONTINUE
      DO 1560 I=NN1,NN2
      II=I-NN1+1
      DO 1560 J=NM1,NM2
      JJ=J-NM1+1
 1560 P(I,J)=PP(KP,II,JJ)
 1562 CONTINUE
      IF(IPRINT.LT.6)GO TO 1565
      WRITE(IW,900)KP,NSY,NSP
      NNP=NN2-NN1+1
      NMP=NM2-NM1+1
      CALL MPRS(P,MN,MN,NNP,NMP,0,0,4HPP   )
 1565 CONTINUE
C
C     FORM Q MATRIX (UI/U)
C
      KYIN=0
      NN1=1
      NN2=0
      DO 1600 N=1,KR
      KYIN=KYIN+NNU(N)
      IF(N.GT.1)NN1=NN1+NNU(N-1)
      NN2=NN2+NNU(N)
      NSY1=NSYS(N)
      DO 1570 KQ=1,MQ
      IF(NSQ(KQ).EQ.NSY1)GO TO 1575
 1570 CONTINUE
      GO TO 1600
 1575 CONTINUE
      DO 1580 I=NN1,NN2
      II=I-NN1+1
      DO 1580 J=1,NUM
 1580 Q(I,J)=QQ(KQ,II,J)
 1600 CONTINUE
      IF(IPRINT.NE.6)GO TO 1610
      CALL MPRS(P,MN,MN,KYIN,KYOUT,T,4HPR   )
      CALL MPRS(Q,MN,NUM,KYIN,NUM,T,4HQR   )
      CALL MPRS(R,NRM,MN,NRM,KYOUT,T,4HRR   )
      CALL MPRS(S,NRM,NUM,NRM,NUM,T,4HSR   )
 1610 CONTINUE
C
C     CALCULATE NR AND NU BY USING Q, R AND S MATRICES
C
      DO 1660 J=1,NUM
      DO 1620 I=1,KYIN
      IF(Q(I,J).NE.0.0)GO TO 1660
 1620 CONTINUE
      DO 1640 I=1,NRM
      IF(S(I,J).NE.0.0)GO TO 1660
 1640 CONTINUE
      NU=J-1
      GO TO 1680
 1660 CONTINUE
      NU=NUM
 1680 CONTINUE
      IF(NU.EQ.0)GO TO 1780
      DO 1740 I=1,NRM
      DO 1730 J=1,KYOUT
```

Figure 32. Subroutine SIMK Program Listing (Continued)

```
      IF(R(I,J).NE.0.0)GO TO 1740
 1700 CONTINUE
      DO 1720 J=1,NU
      IF(S(I,J).NE.0.0)GO TO 1740
 1720 CONTINUE
      NR=I-1
      GO TO 1760
 1740 CONTINUE
      NR=NRM
 1760 CONTINUE
      IF(NR.GT.0)GO TO 1820
C
C     PRINT ERROR MESSEGE
C
 1780 CONTINUE
      WRITE(IW,1800)
 1800 FORMAT(1H1,//,1X,35HINTERCONNECTION SPECIFICATION ERROR)
      STOP 111
C
C     CALCULATE NX AND NY

 1820 CONTINUE
      NX=0
      DO 1840 N=1,KR
      NX=NX+NNX(N)
 1840 CONTINUE
      NY=KYIN+KYOUT
      IF((IPRINT.NE.3).AND.(IPRINT.LT.5))GO TO 1880
      WRITE(IW,1860)
 1860 FORMAT(//,20X,28H*** INTERCONNECTION DATA ***,//)
      CALL MPRS(P,MN,MN,KYIN,KYOUT,T,4HP   )
      CALL MPRS(Q,MN,NUM,KYIN,NU,T,4HQ   )
      CALL MPRS(R,NRM,MN,NR,KYOUT,T,4HR   )
      CALL MPRS(S,NRM,NUM,NR,NU,T,4HS   )
C
C     CALCULATE NSA,NRA AND NUA
C
 1880 CONTINUE
      NXA=NX-NX1
      NRA=NR-NR1
      NUA=NU-NU1
C
C     WRITE INTERCONNECTION DATA ON SCRATCH FILE FOR NAMEL
C     TO FORM NAME LIST DATA
C
      REWIND JS
      IF(IPRINT.EQ.6)WRITE(IW,1890)
 1890 FORMAT(/,1X,30HDATA ON SCRATCH FILE FOR NAMEL,/)
C
C     CALCULATE AND WRITE DATA TO FORM NAME LIST FOR OUTPUTS
C
      CARD(1)=HOUTP
      WRITE(JS,1060)CARD
      IF(IPRINT.EQ.6)WRITE(IW,2000)CARD
 2000 FORMAT(1X,20A4)
      NNRK=
      NNRKP=1
      DO 2100 K=1,KR
      NNRK=NNRK+NNR(K)
      IF(K.GT.1)NNRKP=NNRKP+NNR(K-1)
      DO 2100 I=1,NR
      DO 2100 J=NNRKP,NNRK
      IF(R(I,J).EQ.0.0)GO TO 2100
C     DO 2020 II=1,NR
C     IF(II.EQ.I)GO TO 2020
C     IF(R(II,J).NE.0.0)GO TO 2100
```

Figure 32. Subroutine SIMK Program Listing (Continued)

```
C2020 CONTINUE
      DO 2040 JJ=NNRKP,NNRK
      IF(JJ.EQ.J)GO TO 2040
      IF(R(I,JJ).NE.0.0)GO TO 2100
 2040 CONTINUE
      NNRKK=0
      NNRKKP=1
      DO 2070 KK=1,KB
      NNRKK=NNRKK+NNR(KK)
      IF(KK.GT.1)NNRKKP=NNRKKP+NNR(KK-1)
      IF(KK.EQ.K)GO TO 2070
      DO 2060 JJ=NNRKKP,NNRKK
      IF(R(I,JJ).NE.0.0)GO TO 2100
 2060 CONTINUE
 2070 CONTINUE
      JJJ=J-NNRKP+1
      WRITE(JS,2080)I,K,JJJ
 2080 FORMAT(3I2)
      IF(IPRINT.EQ.6)WRITE(IW,2090)I,K,JJJ
 2090 FORMAT(1X,3(I2,1X))
 2100 CONTINUE
      I=-1
      WRITE(JS,2080)I
      IF(IPRINT.EQ.6)WRITE(IW,2090)I
C
C     CALCULATE AND WRITE DATA TO FORM NAME LIST FOR INPUTS
C
      CARD(1)=HINPU
      WRITE(JS,1060)CARD
      IF(IPRINT.EQ.6)WRITE(IW,2000)CARD
      NNUK=0
      NNUKP=1
      DO 2200 K=1,KB
      NNUK=NNUK+NNU(K)
      IF(K.GT.1)NNUKP=NNUKP+NNU(K-1)
      DO 2200 I=NNUKP,NNUK
      DO 2200 J=1,NU
      IF(Q(I,J).EQ.0.0)GO TO 2200
      DO 2120 II=NNUKP,NNUK
      IF(II.EQ.I)GO TO 2120
      IF(Q(II,J).NE.0.0)GO TO 2200
 2120 CONTINUE
      DO 2140 JJ=1,NU
      IF(JJ.EQ.J)GO TO 2140
      IF(Q(I,JJ).NE.0.0)GO TO 2200
 2140 CONTINUE
      NNUKK=0
      NNUKKP=1
      DO 2170 KK=1,KB
      NNUKK=NNUKK+NNU(KK)
      IF(KK.GT.1)NNUKKP=NNUKKP+NNU(KK-1)
      IF(KK.EQ.K)GO TO 2170
      DO 2160 II=NNUKKP,NNUKK
      IF(Q(II,J).NE.0.0)GO TO 2200
 2160 CONTINUE
 2170 CONTINUE
      III=I-NNUKP+1
      WRITE(JS,2080)J,K,III
      IF(IPRINT.EQ.6)WRITE(IW,2090)J,K,III
 2200 CONTINUE
      J=-1
      WRITE(JS,2080)J
      IF(IPRINT.EQ.6)WRITE(IW,2090)J
      CARD(1)=HEND
      WRITE(JS,1060)CARD
      IF(IPRINT.EQ.6)WRITE(IW,2000)CARD
```

Figure 32. Subroutine SIMK Program Listing (Continued)

```
      RETURN
C
C     COMPUTE SUBSYSTEM STATES   XDOT(N)=AN*XN +BN*UN
C
100   CONTINUE
      II=0
      DO 251 N=1,KB
      NNXN=NNX(N)
      DO 200 I=1,NNXN
      II=II+1
      XDOTL(II)=0.0
      NNUN=NNU(N)
      DO 201 J=1,NNUN
201   XDOTL(II)=XDOTL(II)+BT(I,J,N)*UI(J,N)
      DO 200 J=1,NNXN
200   XDOTL(II)=XDOTL(II)+AT(I,J,N)*X(J,N)
251   CONTINUE
C
C     INTERCONNECTION EQUATIONS
C
C     INTERNAL OUTPUTS RIN=CN*XN+DN*UN
C
      II=0
      DO 35  N=1,KB
      NNRN=NNR(N)
      DO 30  I=1,NNRN
      II=II+1
      YL(II)=0.0
      NNXN=NNX(N)
      DO 301 J=1,NNXN
301   YL(II)=YL(II)+CT(I,J,N)*X(J,N)
      NNUN=NNU(N)
      DO 300  J=1,NNUN
300   YL(II)=YL(II)+DT(I,J,N)*UI(J,N)
350   CONTINUE
C
C     INTERNAL INPUTS
C
      J=0
      DO 22  N=1,KB
      NNRN=NNR(N)
      DO 22  I=1,NNRN
      J=J+1
  220 RIN(J)=RI(I,N)
      DO 24  I=1,KYIN
      II=II+1
      YL(II)=0.0
      DO 23  J=1,KYOUT
  230 YL(II)=YL(II)+P(I,J)*RIN(J)
      DO 24  J=1,NU
  240 YL(II)=YL(II)+Q(I,J)*U(J)
C
C     EXTERNAL RESPONSE EQUATIONS
C
      II=0
      DO 28  I=1,NR
      II=II+1
      RL(II)=0.0
      DO 27  J=1,KYOUT
  270 RL(II)=RL(II)+R(I,J)*RIN(J)
      DO 28  J=1,NU
  280 RL(II)=RL(II)+S(I,J)*U(J)
      RETURN
      END
```

Figure 32. Subroutine SIMK Program Listing (Concluded)

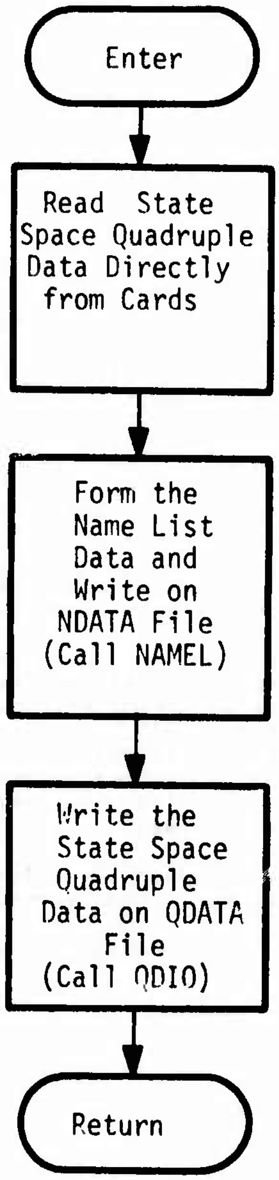

Figure 33. Subroutine QUADK Flow Chart

```
      SUBROUTINE QUADK(A,B,C,D,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     1NNI,VNI,DESI,UNITI,DESSS,UNITSS,DESOO,UNITOO,DESII,UNITII,
     2NXX,NRR,NUU,NXM,NRM,NUM,MB,MB)
C
C     PURPOSE - TO READ DIRECTLY THE STATE MODEL
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        INPT
C        ODIO
C        NAMEL
C        ZERO
C
C     ARGUMENTS LIST
C        DESSS                  STATE DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C        UNITSS                 STATE UNIT ARRAY FOR ALL SUBSYSTEMS
C        DESOO                  OUTPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C        UNITOO                 OUTPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C        DESII                  INPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C        UNITII                 INPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C        NXX                    NO OF STATE ARRAY FOR ALL SUBSYSTEMS
C        NRR                    NO OF OUTPUT ARRAY FOR ALL SUBSYSTEMS
C        NUU                    NO OF INPUT ARRAY FOR ALL SUBSYSTEMS
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB)
      DIMENSION DESOO(NRM,10,MB),UNITOO(NRM,4,MB)
      DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)
      DIMENSION NXX(MB),NRR(MB),NUU(MB)
      DIMENSION CARD(20)
      DATA HC,HSAMP,HXDOT,HSX/1HC,4HSAMP,4HXDOT,4H/X   /
      DATA HSU,HRSX,HRSU,HEND/4H/U   ,4HR/X ,4HR/U ,4HEND /
      T=0.0 $ NX=0 $ NR=0 $ NU=0
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0
      NXA=0 $ NRA=0 $ NUA=0
      CALL ZERO(A,NXM,NXM)
      CALL ZERO(B,NXM,NUM)
      CALL ZERO(C,NRM,NXM)
      CALL ZERO(D,NRM,NUM)
C
C     READ QUADRUPLE DATA FROM CARDS
C
   45 CONTINUE
  100 CONTINUE
      READ(IR,120)CARD
  120 FORMAT(20A4)
      DECODE(4,140,CARD(1))CC,DUMMY
  140 FORMAT(A1,A3)
      IF(CC.EQ.HC)GO TO 100
      IF(CARD(1).NE.HSAMP)GO TO 200
      ENCODE(10,160,TW)CARD(4),CARD(5),CARD(6)
  160 FORMAT(A4,A4,A2)
      DECODE(10,180,TW)T
  180 FORMAT(F10.6)
  200 CONTINUE
```

Figure 34. Subroutine QUADK Program Listing

```
      IF((CARD(1).NE.HXDOT).OR.(CARD(2).NE.HSX))GO TO 220
C
C     READ A MATRIX (XDOT/X)
C
      DECODE(4,210,CARD(4))NX,DUMMY
  210 FORMAT(I3,A1)
      CALL INPT(A,NXM,NXM)
      GO TO 100
  220 CONTINUE
      IF((CARD(1).NE.HXDOT).OR.(CARD(2).NE.HSU))GO TO 240
C
C     READ B MATRIX (XDOT/U)
C
      DECODE(4,210,CARD(4))NX,DUMMY
      DECODE(4,230,CARD(5))DUMMY,NU
  230 FORMAT(A1,I3)
      CALL INPT(B,NXM,NUM)
      GO TO 100
  240 CONTINUE
      IF(CARD(1).NE.HRSX)GO TO 260
C
C     READ C MATRIX (R/X)
C
      DECODE(4,210,CARD(4))NR,DUMMY
      DECODE(4,230,CARD(5))DUMMY,NX
      CALL INPT(C,NRM,NXM)
      GO TO 100
  260 CONTINUE
      IF(CARD(1).NE.HRSU)GO TO 280
C
C     READ D MATRIX (R/U)
C
      DECODE(4,210,CARD(4))NR,DUMMY
      DECODE(4,230,CARD(5))DUMMY,NU
      CALL INPT(D,NRM,NUM)
      GO TO 100
  280 CONTINUE
      IF(CARD(1).NE.HEND)GO TO 320
C
C     READ AND UPDATE NAME LIST DATA
C
      NFLAG=0
      CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
     1DESI,UNITI,DESSS,UNITSS,DESOO,UNITOO,DESII,UNITII,NXX,NRR,NUU,
     2NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)
C
C     WRITE QUADRUPLE DATA ON FILE QDATA
C
      IO=0
      MFLAG=2
      NXA=NX $ NRA=NR $ NUA=NU
      CALL QDIO(A,B,C,D,DESSS,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IO,IPRINT,IW,JQ,HEAD,MARK,
     2LOCATE,NULL,INSERT,MFLAG)
      RETURN
C
C     PRINT ERROR MESSEGE
C
  320 CONTINUE
      WRITE(IW,340)
  340 FORMAT(1H1,//,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)
      STOP 111
      END
```

Figure 34. Subroutine QUADK Program Listing (Concluded)

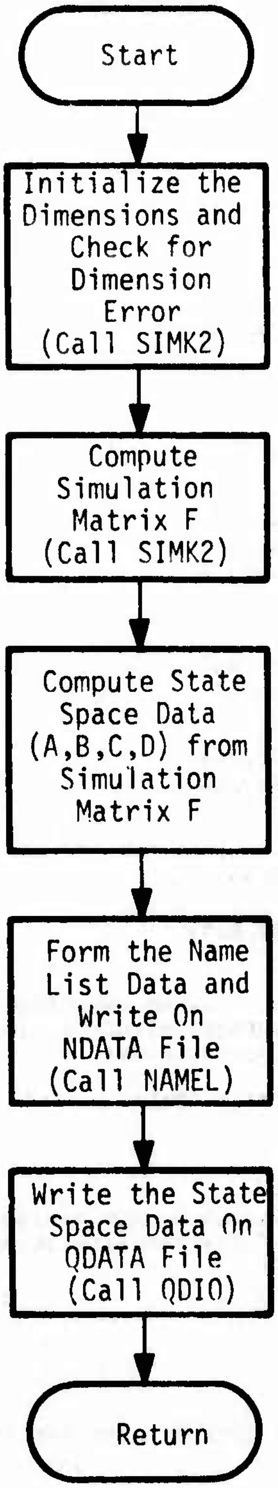

Figure 35. Subroutine STAMK4 Flow Chart

```
      SUBROUTINE STAMK4(V,W,F,U,A,B,C,D,NNS,VNS,DESS,UNITS,
     1NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,MAXN,MAXM,
     2NXM,NRM,NUM,NYM,MB,MS1,MS2,MS3,MS4,NB)
C
C     PURPOSE - TO OBTAIN STATE SPACE MODEL FROM USER WRITTEN
C     SIMULATION EQUATION SUBROUTINE SIMK2
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DERRM
C        MPRS
C        QDIO
C        TDINVR
C        DERRMS
C        NAMEL
C        SIMK2
C
C     ARGUMENTS LIST
C        V                      V ARRAY FOR COMPUTING SIMULATION MATRIX
C        W                      W ARRAY FOR COMPUTING SIMULATION MATRIX
C        F                      SIMULATION MATRIX
C        U                      ARRAY FOR EXTERNAL INPUTS
C        A           IN/OUT     STATE TRANSITION MATRIX
C        B           IN/OUT     CONTROL INPUT MATRIX
C        C           IN/OUT     STATE OUTPUT MATRIX
C        D           IN/OUT     CONTROL OUTPUT MATRIX
C        NNS         IN/OUT     NUMBER ARRAY FOR STATE
C        VNS         IN/OUT     VARIABLE NAME ARRAY FOR STATE
C        DESS        IN/OUT     DESCRIPTION ARRAY FOR STATE
C        UNITS       IN/OUT     UNIT ARRAY FOR STATE
C        NNO         IN/OUT     NUMBER ARRAY FOR OUTPUT
C        VNO         IN/OUT     VARIABLE NAME ARRAY FOR OUTPUT
C        DESO        IN/OUT     DESCRIPTION ARRAY FOR OUTPUT
C        UNITO       IN/OUT     UNIT ARRAY FOR OUTPUT
C        NNI         IN/OUT     NUMBER ARRAY FOR INPUT
C        VNI         IN/OUT     VARIABLE NAME ARRAY FOR INPUT
C        DESI        IN/OUT     DESCRIPTION ARRAY FOR INPUT
C        UNITI       IN/OUT     UNIT ARRAY FOR INPUT
C        MAXN        INPUT      MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C        MAXM        INPUT      MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C        NXM         INPUT      MAXIMUM NUMBER OF STATES
C        NRM         INPUT      MAXIMUM NUMBER OF OUTPUTS
C        NUM         INPUT      MAXIMUM NUMBER OF INPUTS
C        NYM         INPUT      MAXIMUM DIMENSION FOR INTERCONN EQUATIONS
C        MB          INPUT      MAXIMUM NO OF SUBSYSTEMS FOR COMBINING
C        MS1         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C        MS2         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C        MS3         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C        MS4         INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C        NB          INPUT      MAXIMUM SYSTEM NO - IMPLICIT MODEL
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
      DIMENSION U(NUM)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      COMMON /SC1/ S1(1)
C     DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB)
```

Figure 36. Subroutine STAMK4 Program Listing

```
C     DIMENSION DESOO(NRM,1),MB),UNITOO(NRM,4,MB)                       STAMK465
C     DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)                       STAMK466
C     DIMENSION NXX(MB),NRR(MB),NUU(MB)                                 STAMK467
      L1=1 $ L2=L1+NXM*MB*10 $ L3=L2+NXM*MB*4 $ L4=L3+NRM*MB*10         STAMK468
      L5=L4+NRM*MB*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MB*4                 STAMK469
      L8=L7+MB $ L9=L8+MB $ L10=L9+MB                                   STAMK470
      IF(L1 .GT.MS1)                                                    STAMK471
     1CALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,4,0,4HSTAM,4HK4   ,IW) STAMK472
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0                     STAMK473
      NXA=0 $ NRA=0 $ NUA=0                                             STAMK474
      EPSF=1.0E-30 $ T=0.0 $ NFLAG=0                                    STAMK475
C                                                                       STAMK476
C     INITIALIZING CALL TO SUBROUTINE SIMK2                             STAMK477
C                                                                       STAMK478
      INIT=                                                             STAMK479
      NX=0 $ NY=0 $ NR=0 $ NU=0                                         STAMK480
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                             STAMK481
      CALL SIMK2(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),             STAMK482
     1NX,NY,NR,NU,INIT,T)                                               STAMK483
C                                                                       STAMK484
C     CHECK FOR DIMENSION ERROR                                         STAMK485
C                                                                       STAMK486
      INIT = 1                                                          STAMK487
      M=2*NX+NY+NU                                                      STAMK488
      N=NX+NY+NR                                                        STAMK489
      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))      STAMK490
     1CALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,1,0,4HSTAM,4HK1   ,IW)    STAMK491
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                             STAMK492
      DO 101 J=1,M                                                      STAMK493
101   W(J)=0.                                                           STAMK494
      DO 501 J=1,M                                                      STAMK495
      W(J)=1.                                                           STAMK496
      CALL SIMK2(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),             STAMK497
     1NX,NY,NR,NU,INIT,T)                                               STAMK498
      W(J)=0.0                                                          STAMK499
      DO 501 I=1,N                                                      STAMK100
501   F(I,J)=V(I)                                                       STAMK101
C                                                                       STAMK102
C     COMPUTE THE SIMULATION MATRIX                                     STAMK103
C                                                                       STAMK104
      NV=NX+NY                                                          STAMK105
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSTM )                STAMK106
      DO 51 I=1,NV                                                      STAMK107
      DO 52 J=1,NV                                                      STAMK108
   52 F(I,J)=-F(I,J)                                                    STAMK109
   51 F(I,I)=F(I,I)+1.                                                  STAMK110
      CALL TDINVR(ISOL,IDSOL,NV,-M,F,MAXN,KDUM,DET)                     STAMK111
      IB=NV+1                                                           STAMK112
      IE=NV+NR                                                          STAMK113
      JB=IB                                                             STAMK114
      JE=M                                                              STAMK115
      DO 53 I=IB,IE                                                     STAMK116
      DO 53 J=JB,JE                                                     STAMK117
      DO 53 K=1,NV                                                      STAMK118
   53 F(I,J)=F(I,J)+F(I,K)*F(K,J)                                       STAMK119
      DO 53  I=1,IE                                                     STAMK120
      DO 530 J=1,JE                                                     STAMK121
      IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0                              STAMK122
  530 CONTINUE                                                          STAMK123
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSTMI)                STAMK124
C                                                                       STAMK125
C     FORM A,B,C,D MATRICES                                             STAMK126
C                                                                       STAMK127
      J1=NV+1                                                           STAMK128
      J2=NV+NX                                                          STAMK129
      J3=J1+NX                                                          STAMK130
```

Figure 36. Subroutine STAMK4 Program Listing (Continued)

```
      J4=J2+NU
      I1=NV+1
      I2=NV+NR
      DO 6001 I=1,NX
      DO 6001 J=J1,J2
      JJ=J-J1+1
 6001 A(I,JJ)=F(I,J)
      DO 6002 I=1,NX
      DO 6002 J=J3,J4
      JJ=J-J3+1
 6002 B(I,JJ)=F(I,J)
      DO 6003 I=I1,I2
      II=I-I1+1
      DO 6003 J=J1,J2
      JJ=J-J1+1
 6003  C(II,JJ)=F(I,J)
      DO 6004 I=I1,I2
      II=I-I1+1
      DO 6004 J=J3,J4
      JJ=J-J3+1
 6004 D(II,JJ)=F(I,J)
C
C     READ AND UPDATE NAME LIST DATA
C
      KB=NMAX
      CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
     1DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
     2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)
C
C     WRITE QUADRUPLE DATA ON FILE QDATA
C
      IQ=0
      MFLAG=2
      NXA=NX $ NRA=NR $ NUA=NU
      CALL QDIO(A,B,C,D,A,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,HEAD,MARK,
     2LOCATE,NULL,INSERT,MFLAG)
      RETURN
      END
```

Figure 36. Subroutine STAMK4 Program Listing (Concluded)

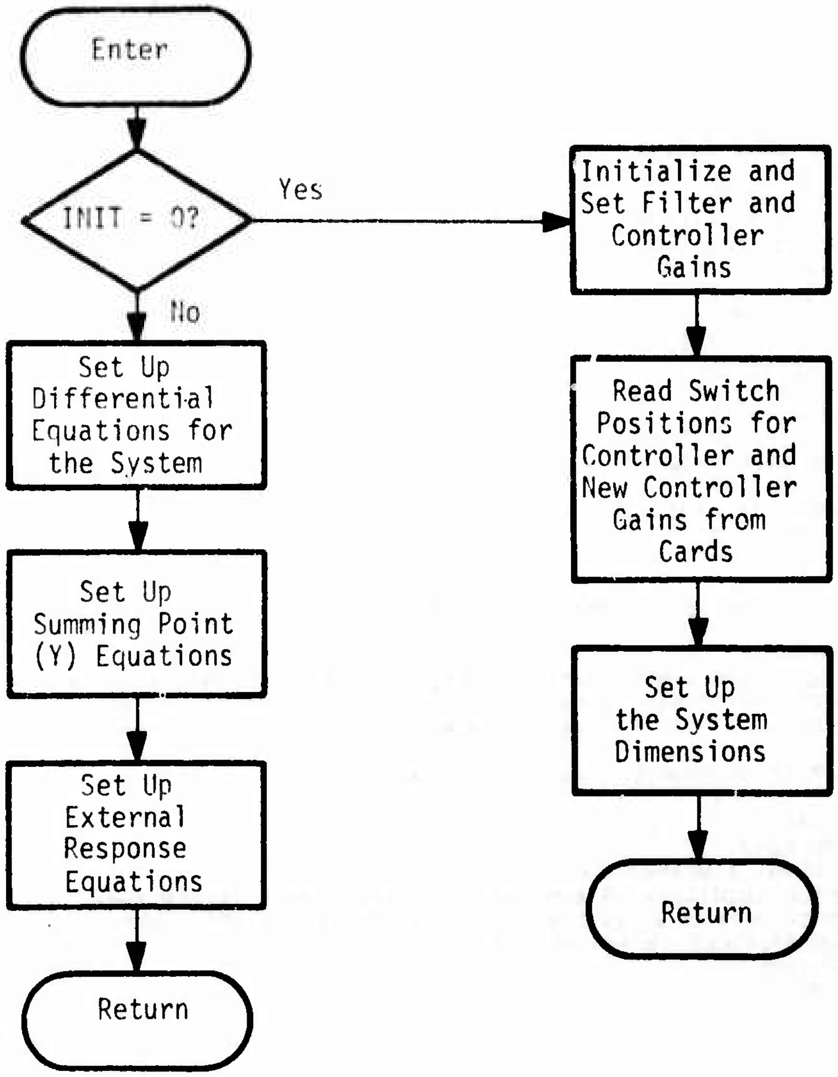

Figure 37. Subroutine SIMK2 Flow Chart

```
      SUBROUTINE SIMK2(XDOT,Y,X,U,XDOTL,YL,RL,NX,NY,NR,NU,INIT,T)
C
C     PURPOSE - TO IMPLEMENT SIMULATION EQUATIONS FOR CSA CONTROLLER
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        XDOT                   ARRAY FOR STATE DERIVATIVES
C        Y                      ARRAY FOR Y EQUATIONS
C        X                      ARRAY FOR STATES
C        U                      ARRAY FOR EXTERNAL INPUTS
C        XDOTL       OUTPUT     ARRAY FOR DERIVATIVE OF STATE
C        YL          OUTPUT     ARRAY FOR Y EQUATION VARIABLES
C        RL          OUTPUT     ARRAY FOR EXTERNAL RESPONSE VARIABLES
C        NX          OUTPUT     NUMBER OF STATES
C        NY          OUTPUT     NUMBER OF Y EQUATIONS
C        NR          OUTPUT     NUMBER OF OUTPUTS
C        NU          OUTPUT     NUMBER OF INPUTS
C        INIT        INPUT      INITIAL MODE FLAG
C        T           OUTPUT     SAMPLE TIME
C
      DIMENSION XDOT(NX),Y(NY),X(NX),U(NU),XDOTL(NX),YL(NY),RL(NR)
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      DIMENSION CARD(20)
      REAL KM1,KM2,KAF,KQ,KP,KNF,MLC1,MLC2
      DATA HENDB,HWITC,HAINB/4HEND ,4HWITC,4HAIN /
      DATA HMLC1,HMLC2,HSASB,HALDC/4HMLC1,4HMLC2,4HSAS ,4HALDC/
      DATA HKM1B,HKM2B,HKAFB,HKQBB/4HKM1 ,4HKM2 ,4HKAF ,4HKQ  /
      DATA HKPBB,HKNFB/4HKP  ,4HKNF /
C
C     CHECK IF INITIALIZATION MODE
C
      IF(INIT.NE.0) GO TO 100
C
C     SET FILTER GAINS
C
      AP=-.1 $ BP=.22361E-03
      ANF=-6.0
      AF=-.02
      AM1=-.01
      AM2=-.01
      AHF=-1. $ BHF=-1.
      ATF=-4.0 $ BTF=4.0
C
C     SET CONTROLLER SWITCHES
C
      SAS=0.0 $ ALDCS=0.0 $ MLC1=0.0 $ MLC2=0.0
C
C     SET CONTROLLER GAINS
C
      KM1=1.0/0.26
      KM2=1.0/0.05591
      KAF=36.0*0.26
      KQ=0.5
      KP=0.3068
      KNF=-0.09
C
C     READ CONTROLLER SWITCHES ON AND CONTROLLER GAIN VALUES
C
   10 CONTINUE
      READ(IR,20)CARD
   20 FORMAT(20A4)
      IF(CARD(1).EQ.HENDB)GO TO 80
```

Figure 38. Subroutine SIMK2 Program Listing

```
      IF(CARD(4).NE.HWITC)GO TO 40
C
C     READ CONTROLLER SWITCHES ON
C
   30 CONTINUE
      READ(IR,20)CARD
      IF(CARD(1).EQ.HENDR)GO TO 10
      IF(CARD(1).EQ.HMLC1)MLC1=1.0
      IF(CARD(1).EQ.HMLC1)GO TO 30
      IF(CARD(1).EQ.HMLC2)MLC2=1.0
      IF(CARD(1).EQ.HMLC2)GO TO 30
      IF(CARD(1).EQ.HSASR)SAS=1.0
      IF(CARD(1).EQ.HSASR)GO TO 30
      IF(CARD(1).EQ.HALDC)ALDCS=1.0
      IF(CARD(1).EQ.HALDC)GO TO 30
      STOP 111
C
C     READ CONTROLLER GAIN VALUES
C
   40 CONTINUE
      IF(CARD(4).NE.HAINR)STOP 111
   50 CONTINUE
      READ(IR,20)CARD
      IF(CARD(1).EQ.HENDR)GO TO 10
      IF(CARD(1).EQ.HKM1R)READ(IR,60)KM1
   60 FORMAT(E12.6)
      IF(CARD(1).EQ.HKM1R)GO TO 50
      IF(CARD(1).EQ.HKM2R)READ(IR,60)KM2
      IF(CARD(1).EQ.HKM2R)GO TO 50
      IF(CARD(1).EQ.HKAFR)READ(IR,60)KAF
      IF(CARD(1).EQ.HKAFR)GO TO 50
      IF(CARD(1).EQ.HKQBR)READ(IR,60)KQ
      IF(CARD(1).EQ.HKQBR)GO TO 50
      IF(CARD(1).EQ.HKPBR)READ(IR,60)KP
      IF(CARD(1).EQ.HKPBB)GO TO 50
      IF(CARD(1).EQ.HKNFR)READ(IR,60)KNF
      IF(CARD(1).EQ.HKNFB)GO TO 50
      STOP 111
   80 CONTINUE
C
C     SET DIMENSIONS OF SYSTEM
C
      NX=7 $ NR=3 $ NU=9 $ NY=5
C
C     RETURN
C
      RETURN
C
C     SIMULATION EQUATIONS
C
  100 CONTINUE
C
C     DIFFERENTIAL EQUATIONS
C
      XDOTL(1)=AP*X(1)+BP*U(3)
      XDOTL(2)=ANF*X(2)+ALDCS*U(6)
      XDOTL(3)=AM1*X(3)+MLC1*Y(2)
      XDOTL(4)=AF*X(4)+ALDCS*Y(1)
      XDOTL(5)=ATF*X(5)+BTF*Y(3)
      XDOTL(6)=AHF*X(6)+BHF*U(2)
      XDOTL(7)=AM2*X(7)+MLC2*Y(4)
C
C     SUMMING POINT EQUATIONS
C
      YL(1)=KAF*X(2)+ANF*U(4)
      YL(2)=KM1*U(4)-U(9)
```

Figure 38. Subroutine SIMK2 Program Listing (Continued)

```
      YL(3)=KP*X(1)+X(6)+U(2)+KNF*U(7)
      YL(4)=KM2*U(5)-U(9)
      YL(5)=ALDCS*X(5)+X(1)+SAS*KQ*U(8)
C
C     RESPONSE EQUATIONS
C
      RL(1)=U(1)
      RL(2)=Y(5)
      RL(3)=X(1)
C
C     RETURN
C
      RETURN
      END
```

Figure 38. Subroutine SIMK2 Program Listing (Concluded)

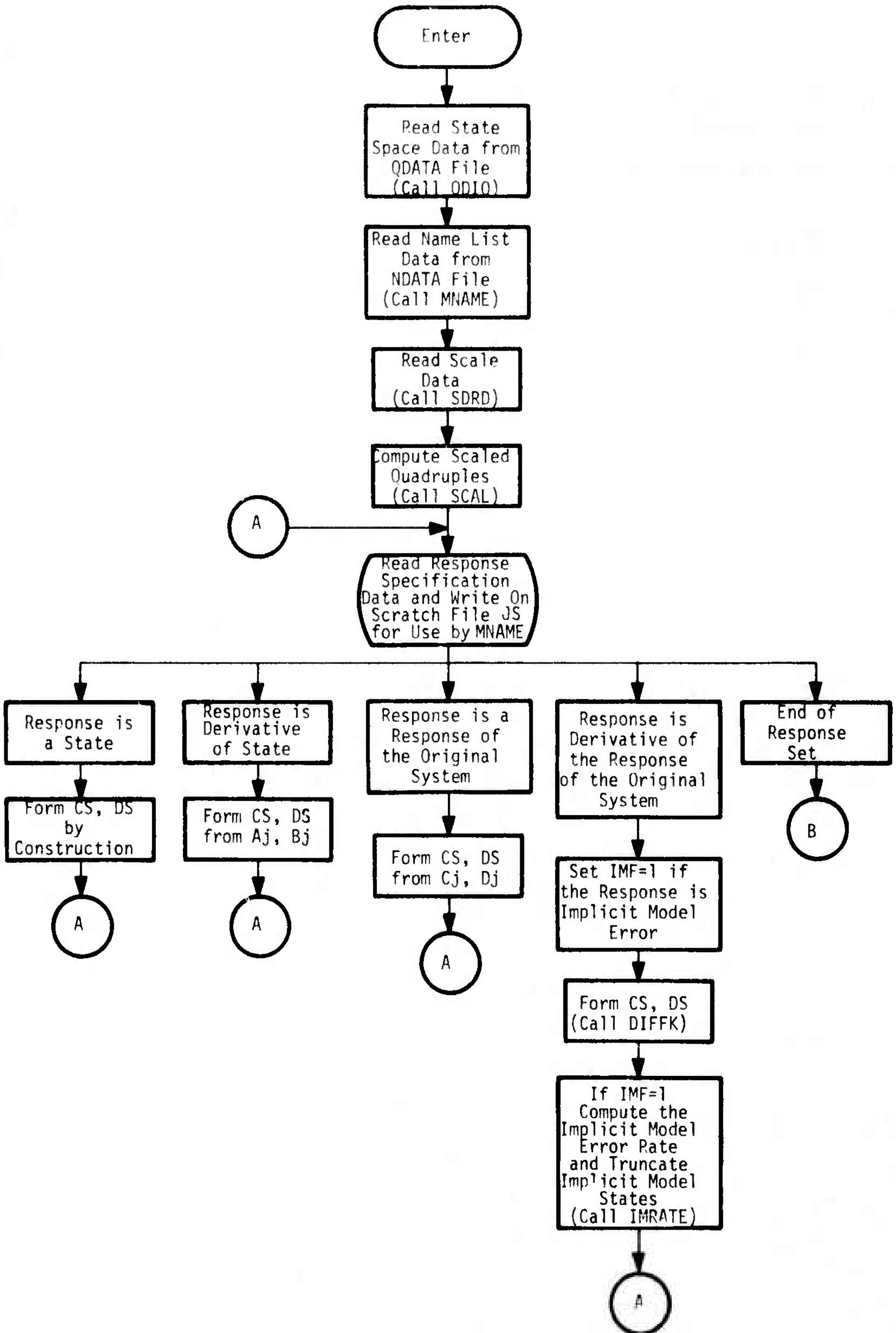

Figure 39. Subroutine CONDK Flow Chart

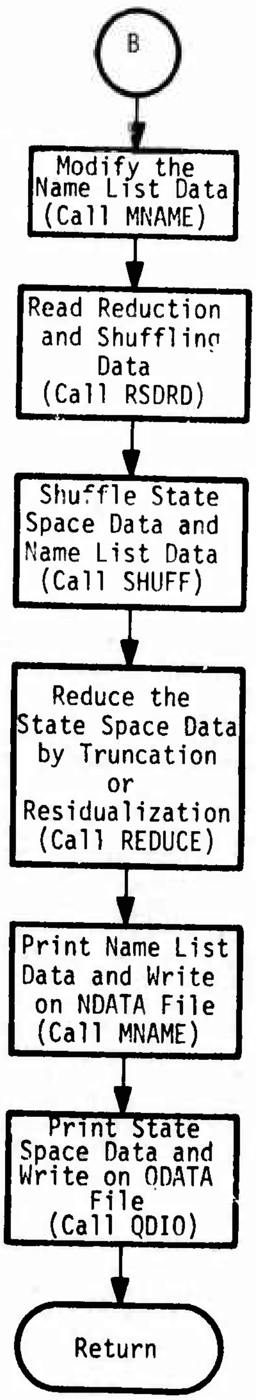

Figure 39. Subroutine CONDK Flow Chart (Concluded)

```
      SUBROUTINE CONDK(A,B,C,D,CM,DM,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,
     1UNITO,NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,
     2UNITNO,NNNI,VNNI,DESNI,UNITNI,DUMMY1,DUMMY2,DUMMY3,ES,ER,NSHUFS,
     3NSHUFO,NSHUFI,CS,DS,CW,DW,IRS,Q,NXM,NRM,NUM,NDM11,NDM12,NDM21,
     4NDM22)
C
C     PURPOSE - TO CONDITION THE STATE SPACE QUADRUPLE DATA
C     (RESPONSE SPECIFICATIONS, TRUNCATION AND RESIDUALIZATION
C     AND SHUFFLING)
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C     NOTE - PREVIOUS NAME OF THE SUBROUTINE IS RESPK
C
C     SUBPROGRAMS CALLED
C        QDIO
C        DEBUG
C        MNAME
C        ERRM
C        SDRD
C        SCAL
C        DIFFK
C        IMPATE
C        RSDRD
C        SHUFF
C        REDUCE
C        HPR
C
C     ARGUMENTS LIST
C        A          IN/OUT    STATE TRANSITION MATRIX
C        B          IN/OUT    CONTROL INPUT MATRIX
C        C          IN/OUT    STATE OUTPUT MATRIX
C        D          IN/OUT    CONTROL OUTPUT MATRIX
C        CM         IN/OUT    MODIFIED STATE OUTPUT MATRIX
C        DM         IN/OUT    MODIFIED CONTROL OUTPUT MATRIX
C        NNS        IN/OUT    OLD NUMBER ARRAY FOR STATE
C        VNS        IN/OUT    OLD VARIABLE NAME ARRAY FOR STATE
C        DESS       IN/OUT    OLD DESCRIPTION ARRAY FOR STATE
C        UNITS      IN/OUT    OLD UNIT ARRAY FOR STATE
C        NNO        IN/OUT    OLD NUMBER ARRAY FOR OUTPUT
C        VNO        IN/OUT    OLD VARIABLE NAME ARRAY FOR OUTPUT
C        DESO       IN/OUT    OLD DESCRIPTION ARRAY FOR OUTPUT
C        UNITO      IN/OUT    OLD UNIT ARRAY FOR OUTPUT
C        NNI        IN/OUT    OLD NUMBER ARRAY FOR INPUT
C        VNI        IN/OUT    OLD VARIABLE NAME ARRAY FOR INPUT
C        DESI       IN/OUT    OLD DESCRIPTION ARRAY FOR INPUT
C        UNITI      IN/OUT    OLD UNIT ARRAY FOR INPUT
C        NNNS       IN/OUT    NEW NUMBER ARRAY FOR STATE
C        VNNS       IN/OUT    NEW VARIABLE NAME ARRAY FOR STATE
C        DESNS      IN/OUT    NEW DESCRIPTION ARRAY FOR STATE
C        UNITNS     IN/OUT    NEW UNIT ARRAY FOR STATE
C        NNNO       IN/OUT    NEW NUMBER ARRAY FOR OUTPUT
C        VNNO       IN/OUT    NEW VARIABLE NAME ARRAY FOR OUTPUT
C        DESNO      IN/OUT    NEW DESCRIPTION ARRAY FOR OUTPUT
C        UNITNO     IN/OUT    NEW UNIT ARRAY FOR OUTPUT
C        NNNI       IN/OUT    NEW NUMBER ARRAY FOR INPUT
C        VNNI       IN/OUT    NEW VARIABLE NAME ARRAY FOR INPUT
C        DESNI      IN/OUT    NEW DESCRIPTION ARRAY FOR INPUT
C        UNITNI     IN/OUT    NEW UNIT ARRAY FOR INPUT
C        DUMMY1               SCRATCH ARRAY
C        DUMMY2               SCRATCH ARRAY
C        DUMMY3               SCRATCH ARRAY
C        ES         OUTPUT    STATE RESIDUALIZATION ERROR MATRIX
C        ER         OUTPUT    OUTPUT RESIDUALIZATION ERROR MATRIX
```

Figure 40. Subroutine CONDK Program Listing

```
C        NSHUFS      INPUT       SHUFFLING ARRAY FOR STATE
C        NSHUFO      INPUT       SHUFFLING ARRAY FOR OUTPUT
C        NSHUFI      INPUT       SHUFFLING ARRAY FOR INPUT
C        CS          IN/OUT      SPECIFIED STATE OUTPUT MATRIX
C        DS          IN/OUT      SPECIFIED CONTROL OUTPUT MATRIX
C        CW          IN/OUT      IMPLICIT MODEL STATE OUTPUT MATRIX
C        DW          IN/OUT      IMPLICIT MODEL CONTROL OUTPUT MATRIX
C        IRS         IN/OUT      ARRAY FOR DERIVATIVES OF RESPONSES
C        Q           IN/OUT      QUADRATIC WEIGHT MATRIX
C        NXM         INPUT       MAXIMUM NUMBER OF STATES
C        NRM         INPUT       MAXIMUM NUMBER OF OUTPUTS
C        NUM         INPUT       MAXIMUM NUMBER OF INPUTS
C        NDM11       INPUT       MAX ROW DIMENSION FOR SCRATCH ARRAY DUMMY1
C        NDM12       INPUT       MAX COL DIMENSION FOR SCRATCH ARRAY DUMMY1
C        NDM21       INPUT       MAX ROW DIMENSION FOR SCRATCH ARRAY DUMMY2
C        NDM22       INPUT       MAX COL DIMENSION FOR SCRATCH ARRAY DUMMY2
C
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      COMMON /SYS/ SCODE,SDES(5),NSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION CM(NRM,NXM),DM(NRM,NUM)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DIMENSION NNNS(NXM),VNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4)
      DIMENSION NNNO(NRM),VNNO(NRM,2),DESNO(NRM,10),UNITNO(NRM,4)
      DIMENSION NNNI(NUM),VNNI(NUM,2),DESNI(NUM,10),UNITNI(NUM,4)
      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22),DUMMY3(NUM)
      DIMENSION ES(NXM,NUM),ER(NRM,NUM)
      DIMENSION NSHUFS(NXM),NSHUFO(NRM),NSHUFI(NUM)
      DIMENSION CS(NRM,NXM),DS(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
      DIMENSION IRS(NRM),Q(NRM,NRM)
      DIMENSION CARD(20)
      EQUIVALENCE (NU1,NCL),(NU2,NGT),(NU3,NCD),(NR1,NDR),(NR2,NPR),
     1(NR3,NSR)
      DATA HENDB,HRBBB,HRDOT,HXDOT/4HEND ,4HR   ,4HRDOT,4HXDOT/
      DATA HXBBB,HSCAL/4HX   ,4HSCAL/
      DATA HCONS,HSELE/4HCONS,4HSELE/
      DATA HSIGN,HRFOR,HENSO/4HSIGN,4HRFOR,4HENSO/
      DATA HONTR,HUSTB,HOMMA/4HONTR,4HUST ,4HOMMA/
C
C     READ QUADRUPLE DATA
C
      IQ=0
      NFLAG=1
      CALL QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,PHEAD,MARK,
     2LOCATE,NULL,INSERT,NFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HRESP,4HK   ,5,0,IW)
C
C     READ NAME LIST DATA
C
      MFLAG=1
      CALL MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,
     2UNITNO,NNNI,VNNI,DESNI,UNITNI,NX,NR,NU,NXM,NRM,NUM,
     3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HRESP,4HK   ,5,0,IW)
      REWIND JS
      READ(IR,140)CARD
      IF(CARD(1).EQ.HENDB)GO TO 110
      IF(CARD(1).NE.HSCAL)CALL ERRM(1,4HRESP,4HK   ,5,0,IW)
C
C     READ SCALE AND NEW UNIT DATA
C
```

Figure 40. Subroutine CONDK Program Listing (Continued)

```
      CALL SORD(DUMMY1,UNITNS,UNITS,DUMMY2,UNITNO,UNITO,DUMMY3,UNITNI,
     1UNITI,NX,NR,NU,NXM,NRM,NUM,IR,IW,IPRINT)
C
C     COMPUTE SCALED QUADRUPLES
C
      CALL SCAL(A,B,C,D,DUMMY1,DUMMY2,DUMMY3,NX,NR,NU,NXM,NRM,NUM)
C
C     READ RESPONSE SPECIFICATION DATA AND WRITE IT
C     ON A SCRATCH FILE JS FOR USE BY SUBROUTINE MNAME
C
  110 CONTINUE
      NXO=NX $ NRO=NR
      IFLAG=0
      IRR=0 $ K=0 $ NDR=0 $ NPR=0 $ NSR=0
      JU=0 $ NCL=0 $ NGT=0 $ NCD=0
      IRESP=0
  120 CONTINUE
      READ(IR,140)CARD
  140 FORMAT(20A4)
      WRITE(JS,140)CARD
      IF(CARD(1).EQ.HENDR)GO TO 570
      IRESP=1
      IF((CARD(1).NE.HCONS).AND.(CARD(1).NE.HSELE))
     1CALL ERRM(2,4HRESP,4HK    ,5,0,IW)
      IF((CARD(3).NE.HONTR).AND.(CARD(3).NE.HUSTB).AND.(CARD(3).NE.
     1HOMMA))GO TO 240
C
C     READ INPUT SPECIFICATION AND MODIFY B AND D MATRICES
C
      IF(CARD(3).EQ.HONTR)IUU=1
      IF(CARD(3).EQ.HUSTB)IUU=2
      IF(CARD(3).EQ.HOMMA)IUU=3
  160 CONTINUE
      READ(IR,140)CARD
      WRITE(JS,140)CARD
      IF(CARD(1).EQ.HENDB)GO TO 220
      JU=JU+1
      DECODE(4,340,CARD(2))D1,K,D2
      DO 180 I=1,NX
  180 DUMMY1(I,JU)=B(I,K)
      DO 200 I=1,NR
  200 DUMMY2(I,JU)=D(I,K)
      GO TO 160
  220 CONTINUE
      IF(IUU.EQ.1)NCL=JU
      IF(IUU.EQ.2)NGT=JU-NCL
      IF(IUU.EQ.3)NCD=JU-NCL-NGT
      GO TO 120
  240 CONTINUE
      IF(IFLAG.EQ.1)GO TO 300
      NUN=NCL+NGT+NCD
      IF(NUN.EQ.0)GO TO 300
      NU=NUN
      DO 260 I=1,NX
      DO 260 J=1,NU
  260 B(I,J)=DUMMY1(I,J)
      DO 280 I=1,NR
      DO 280 J=1,NU
  280 D(I,J)=DUMMY2(I,J)
      IFLAG=1
C
C     READ OUTPUT SPECIFICATION AND COMPUTE C AND D MATRICES
C
  300 CONTINUE
      IF(CARD(4).EQ.HSIGN)IRR=1
      IF(CARD(4).EQ.HRFOR)IRR=2
```

Figure 40. Subroutine CONDK Program Listing (Continued)

```
      IF(CARD(3).EQ.HENSO)IRR=3
      IM=0 $ IMF=0 $ J=0
  320 CONTINUE
      READ(IR,140)CARD
      WRITE(JS,140)CARD
      IF(CARD(1).EQ.HENDR)GO TO 550
      J=J+1
      DECODE(4,340,CARD(2))D1,K,D2
  340 FORMAT(A1,I2,A1)
      ID=0 $ IX=0
      IF(CARD(1).EQ.HRBBR)GO TO 360
      ID=0 $ IX=1
      IF(CARD(1).EQ.HXBBR)GO TO 360
      ID=1 $ IX=0
      IF(CARD(1).EQ.HRDOT)GO TO 360
      ID=1 $ IX=1
      IF(CARD(1).EQ.HXDOT)GO TO 360
      CALL ERRM(3,4HRESP,4HK    ,5,0,IW)
  360 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HRESP,4HK    ,5,0,IW)
      IF(ID.EQ.1)GO TO 440
      IF(IX.EQ.1)GO TO 420
      DO 380 L=1,NX
  380 CS(J,L)=C(K,L)
      DO 400 L=1,NU
  400 DS(J,L)=D(K,L)
      GO TO 320
  420 CONTINUE
      DO 425 L=1,NX
  425 CS(J,L)=0.0
      CS(J,K)=1.0
      DO 430 L=1,NU
  430 DS(J,L)=0.0
      GO TO 320
  440 CONTINUE
      IF(IX.EQ.1)GO TO 500
      NRS=1
      IRS(1)=K
      IF(K.LE.NRA)GO TO 450
C
C     DEFINE FLAGS AND RESPONSES FOR IMPLICIT MODEL OPERATION
C
      IMF=1
      IM=IM+1
      DO 443 L=1,NX
  443 CW(IM,L)=C(K,L)
      DO 446 L=1,NU
  446 DW(IM,L)=D(K,L)
  450 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HRESP,4HK    ,5,0,IW)
C
C     COMPUTE DERIVATIVES OF RESPONSES
C
      CALL DIFFK(A,B,C,D,DUMMY1,DUMMY2,NX,NR,NU,
     1NXM,NRM,NUM,NRS,IRS,ID,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
      DO 460 L=1,NX
  460 CS(J,L)=C(NR,L)
      DO 480 L=1,NU
  480 DS(J,L)=D(NR,L)
      NR=NR-1
      GO TO 320
  500 CONTINUE
      DO 520 L=1,NX
  520 CS(J,L)=A(K,L)
      DO 540 L=1,NU
  540 DS(J,L)=B(K,L)
```

Figure 40. Subroutine CONDK Program Listing (Continued)

```
      GO TO 320
  550 CONTINUE
      NR=J
      IF(IMF.EQ.0)GO TO 560
C
C     COMPUTE IMPLICIT MODEL ERROR RATES AND TRUNCATE THE
C     IMPLICIT MODEL
C
      NXR=NX-NXA
      IF(IM.NE.NXR)CALL ERRM(4,4HRESP,4HK    ,5,0,IW)
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HRESP,4HK    ,5,0,IW)
      CALL IMRATE(CS,DS,CW,DW,DUMMY1,DUMMY2,NX,NR,NU,
     1NXM,NRM,NUM,NXA,NRA,NUA,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
  560 CONTINUE
C
C     COMPUTE NEW C AND D MATRICES (OUTPUTS)
C
      NXN=NXA
      K=0
      IF(IRR.EQ.0)NNR=NR
      IF(IRR.EQ.1)NDR=NR
      IF(IRR.EQ.2)NPR=NR
      IF(IRR.EQ.2)K=NDR
      IF(IRR.EQ.3)NSR=NR
      IF(IRR.EQ.3)K=NDR+NPR
      DO 564 I=1,NR
      IK=I+K
      DO 562 J=1,NX
  562 CM(IK,J)=CS(I,J)
      DO 564 J=1,NU
  564 DM(IK,J)=DS(I,J)
      NX=NXO $ NR=NRO
      GO TO 120
  570 CONTINUE
      IF(IRESP.EQ.0)GO TO 595
      IF(IPRINT.EQ.6)CALL DEBUG(6,4HRESP,4HK    ,5,0,IW)
      IF(IRR.EQ.0)NR=NNR
      IF(IRR.EQ.0)GO TO 575
      NX=NXN $ NR=NDR+NPR+NSR
  575 CONTINUE
      DO 590 I=1,NR
      DO 580 J=1,NX
  580 C(I,J)=CM(I,J)
      DO 590 J=1,NU
  590 D(I,J)=DM(I,J)
C
C     MODIFY NAME LIST DATA
C
      MFLAG=2
      CALL MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,
     2UNITNO,NNNI,VNNI,DESNI,UNITNI,NX,NR,NU,NXM,NRM,NUM,
     3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(7,4HRESP,4HK    ,5,0,IW)
  595 CONTINUE
C
C     READ SHUFFLING AND REDUCTION DATA
C
      CALL RSDRD(DUMMY1,NSHUFS,DUMMY2,NSHUFO,DUMMY3,NSHUFI,
     1NX,NR,NU,NXRN,NXN,NXR,NRN,NRR,NRT,NUN,NXM,NRM,NUM,
     2IR,IW,IPRINT,IRSS)
      IF(IRSS.EQ.0)GO TO 600
      IF(IPRINT.EQ.6)CALL DEBUG(8,4HRESP,4HK    ,5,0,IW)
      IF(IPRINT.LT.6)GO TO 598
      WRITE(IW,596)NX,NR,NU,NXM,NRM,NUM,NDM
      WRITE(IW,596)NSHUFS,NSHUFO,NSHUFI
```

Figure 40. Subroutine CONDK Program Listing (Continued)

```
  596 FORMAT(1X,(20I2,1X))
  598 CONTINUE
C
C     SHUFFLE QUADRUPLE DATA AND NAME LIST DATA
C
      CALL SHUFF(A,B,C,D,NNS,VNS,DESNS,UNITNS,NNO,
     1VNO,DESNO,UNITNO,NNI,VNI,DESNI,UNITNI,
     2NSHUFS,NSHUFO,NSHUFI,DUMMY1,DUMMY2,NX,NR,NU,NXM,NRM,NUM,
     3NDM11,NDM12,NDM21,NDM22)
C
C     TRUNCATE THE SYSTEM VARIABLES AS SPECIFIED
C
      NX=NXRN
      NR=NRN
      IF(NRN.EQ.0)NR=NRT+NRR
      NU=NUN
      IF(NXR.LE.0)GO TO 600
C
C     REDUCE THE QUADRUPLE DATA
C
      IF(IPRINT.EQ.5)CALL HPR(HEAD,IW)
      CALL REDUCE(A,B,C,D,DUMMY1,DUMMY2,ES,ER,
     1NX,NR,NU,NXR,NRR,NRT,NXM,NRM,NUM,T,IW,IPRINT,
     2NDM11,NDM12,NDM21,NDM22)
      IF(IPRINT.EQ.6)CALL DEBUG(9,4HRESP,4HK    ,5,0,IW)
C
C     WRITE NAME LIST DATA
C
  600 CONTINUE
      MFLAG=3
      CALL MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,
     2UNITNO,NNNI,VNNI,DESNI,UNITNI,NX,NR,NU,NXM,NRM,NUM,
     3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(10,4HRESP,4HK    ,5,0,IW)
C
C     WRITE QUADRUPLE DATA
C
      IQ=0 $ NFLAG=2
      NXA=NX $ NRA=NR $ NUA=NU
      CALL QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,HEAD,MARK,
     2LOCATE,NULL,INSERT,NFLAG)
      IF(IPRINT.EQ.6)CALL DEBUG(11,4HRESP,4HK    ,5,0,IW)
      RETURN
      END
```

Figure 40. Subroutine CONDK Program Listing (Concluded)

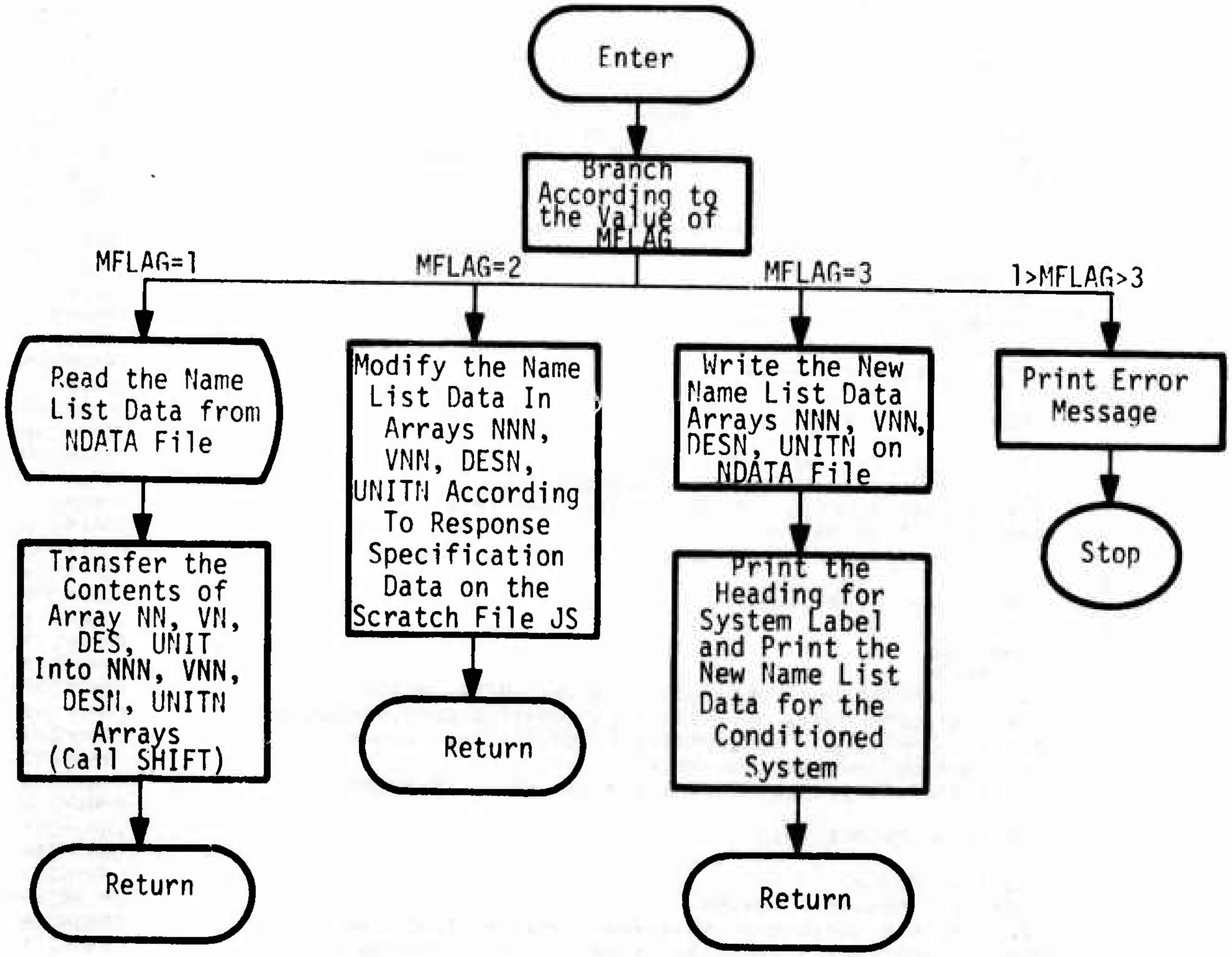

Figure 41. Subroutine MNAME Flow Chart

```
      SUBROUTINE MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
     1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,
     2UNITNO,NNNI,VNNI,DESNI,UNITNI,NXN,NRN,NUN,NXM,NRM,NUM,
     3NCL,NGT,NCD,NDR,NPR,NSR,MFLAG)
C
C     PURPOSE - TO READ, MANIPULATE AND PRINT NAME LIST TABLE
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        FILE
C        ERRM
C        SHIFT
C        DEBUG
C        HPP
C
C     ARGUMENTS LIST
C        NXN        OUTPUT     NUMBER OF STATES
C        NRN        OUTPUT     NUMBER OF OUTPUTS
C        NUN        OUTPUT     NUMBER OF INPUTS
C        NCL        OUTPUT     NO OF CONTROL INPUTS
C        NGT        OUTPUT     NO OF GUST INPUTS
C        NCD        OUTPUT     NO OF COMMAND INPUTS
C        NDR        OUTPUT     NO OF DESIGN OUTPUTS
C        NPR        OUTPUT     NO OF PERFORMANCE OUTPUTS
C        NSR        OUTPUT     NO OF SENSOR OUTPUTS
C        MFLAG      INPUT      CONTROLS ENTRY POINT IN THE SUBROUTINE
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
     1,PHEAD(20)
      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DIMENSION NNNS(NXM),VNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4)
      DIMENSION NNNO(NRM),VNNO(NRM,2),DESNO(NRM,10),UNITNO(NRM,4)
      DIMENSION NNNI(NUM),VNNI(NUM,2),DESNI(NUM,10),UNITNI(NUM,4)
      DIMENSION CARD(20)
      DATA HONTR,HUSTR,HOMMA/4HONTR,4HUST ,4HOMMA/
      DATA HPBBB,HBBBP,HBDES/4H(   ,4H   ),4H DES/
      DATA HBBBB,HENDR/4H    ,4HEND /
      DATA HXP,HRP,HUP,HP/2HX(,2HR(,2HU(,4H)   /
      DATA HDSDT,HOF,HSSEC/4HD/DT,4H OF ,4H/SEC/
      DATA HXBBB,HRBBB,HXDOT,HRDOT/4HX   ,4HR   ,4HXDOT,4HRDOT/
      REWIND JS
C
C     READ THE OLD NAME LIST TABLE OFF THE DATA FILE
C     AND TRANSFER INTO NEW NAME LIST TABLE
C
      IF(MFLAG.NE.1)GO TO 160
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HMNAM,4HE   ,5,0,IW)
      IF(IPRINT.EQ.6)WRITE(IW,150)PHEAD
  150 FORMAT(1X,20A4)
      CALL FILE(JN,LOCATE,PHEAD)
      READ(JN)NX,NR,NU,
     1        (NNS(I),(VNS(I,J),J=1,2),
     2        (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NX),
     3        (NNO(I),(VNO(I,J),J=1,2),
     4        (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NR),
     5        (NNI(I),(VNI(I,J),J=1,2),
     6        (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NU)
      CALL SHIFT(NNS,VNS,DESS,UNITS,NNNS,VNNS,DESNS,UNITNS,NX,NXM,
```

Figure 42. Subroutine MNAME Program Listing

```
     1IW,IPRINT)
      CALL SHIFT(NNO,VNO,DESO,UNITO,NNNO,VNNO,DESNO,UNITNO,NR,NRM,
     1IW,IPRINT)
      CALL SHIFT(NNI,VNI,DESI,UNITI,NNNI,VNNI,DESNI,UNITNI,NU,NUM,
     1IW,IPRINT)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HMNAM,4HE    ,5,0,IW)
      RETURN
C
C     MODIFY THE NAME LIST TABLE FOR INPUT VARIABLES
C     IF THE INPUT SPACE IS EXPANDED
C
  160 CONTINUE
      IF(MFLAG.NE.2)GO TO 760
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HMNAM,4HE    ,5,0,IW)
      IF(NUM.LT.NU)CALL ERRM(1,4HMNAM,4HE    ,6,0,IW)
      IF(NUM.EQ.NU)GO TO 280
      NUS=NU
  180 CONTINUE
      DO 260 I=1,NUS
      II=NU+I
      NNNI(II)=II
      ENCODE(4,200,VNNI(II,1))HUP,II
  200 FORMAT(A2,I2)
      VNNI(II,2)=HP
      DO 220 J=1,4
  220 UNITNI(II,J)=UNITI(I,J)
      DO 240 J=1,10
  240 DESNI(II,J)=DESI(I,J)
  260 CONTINUE
      NU=NU+NUS
      IF(NUM.LT.NU)CALL ERRM(2,4HMNAM,4HE    ,6,0,IW)
      IF(NUM.EQ.NU)GO TO 280
      GO TO 180
C
C     MODIFY THE NAME LIST FOR OUTPUT VARIABLES AND INPUT VARIABLES
C
  280 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HMNAM,4HE    ,5,0,IW)
      IFLAG=0
      J=0 $ IRR=0 $ JU=0 $ IUU=0
  340 CONTINUE
      READ(JS,480)CARD
      IF(CARD(1).EQ.HENDR)RETURN
      IF((CARD(3).NE.HONTR).AND.(CARD(3).NE.HUSTB).AND.(CARD(3).NE.
     1HOMMA))GO TO 460
C
C     OBTAIN NAME LIST DATA FOR SPECIFIED INPUTS
C
  360 CONTINUE
      READ(JS,480)CARD
      IF(CARD(1).EQ.HENDR)GO TO 340
      JU=JU+1
      DECODE(4,500,CARD(2))D1,K,D2
      NNNI(JU)=JU
      NNI(JU)=JU
      ENCODE(4,200,VNNI(JU,1))HUP,JU
      VNNI(JU,2)=HP
      VNI(JU,1)=VNNI(JU,1)
      VNI(JU,2)=VNNI(JU,2)
      DO 380 L=1,10
  380 DESNI(JU,L)=DESI(K,L)
      DO 400 L=1,4
  400 UNITNI(JU,L)=UNITI(K,L)
      GO TO 360
C
C     OBTAIN NAME LIST DATA FOR SPECIFIED OUTPUTS
```

Figure 42. Subroutine MNAME Program Listing (Continued)

```
C
  460 CONTINUE
      READ(JS,480)CARD
  480 FORMAT(20A4)
      IF(CARD(1).EQ.HENDR)GO TO 340
      J=J+1
      DECODE(4,500,CARD(2))ID1,K,ID2
  500 FORMAT(A1,I2,A1)
      ID=0 $ IX=0
      IF(CARD(1).EQ.HRBAR)GO TO 520
      ID=0 $ IX=1
      IF(CARD(1).EQ.HXBAR)GO TO 520
      ID=1 $ IX=0
      IF(CARD(1).EQ.HRDOT)GO TO 520
      ID=1 $ IX=1
      IF(CARD(1).EQ.HXDOT)GO TO 520
      CALL ERRM(3,4HMNAM,4HE   ,6,0,IW)
  520 CONTINUE
      NNNO(J)=J
      NNO(J)=J
      ENCODE(4,200,VNNO(J,1))HRP,J
      VNNO(J,2)=HP
      VNO(J,1)=VNNO(J,1)
      VNO(J,2)=VNNO(J,2)
      IF(ID.EQ.1)GO TO 640
      IF(IX.EQ.1)GO TO 580
      DO 540 L=1,10
  540 DESNO(J,L)=DESO(K,L)
      DO 560 L=1,4
  560 UNITNO(J,L)=UNITO(K,L)
      GO TO 460
  580 CONTINUE
      DO 600 L=1,10
  600 DESNO(J,L)=DESS(K,L)
      DO 620 L=1,4
  620 UNITNO(J,L)=UNITS(K,L)
      GO TO 460
C
C     FORM NAME LIST DATA FOR DERIVATIVES OF SPECIFIED STATES
C     OR OUTPUTS
C
  640 CONTINUE
      DESNO(J,1)=HDSDT
      DESNO(J,2)=HOF
      DESNO(J,3)=HPBAR
      DESNO(J,10)=HRBAR
      UNITNO(J,4)=HSSEC
      IF(IX.EQ.1)GO TO 700
      DO 660 L=1,6
      LL=3+L
  660 DESNO(J,LL)=DESO(K,L)
      DO 680 L=1,3
  680 UNITNO(J,L)=UNITO(K,L)
      GO TO 460
  700 CONTINUE
      DO 720 L=1,6
      LL=3+L
  720 DESNO(J,LL)=DESS(K,L)
      DO 740 L=1,3
  740 UNITNO(J,L)=UNITS(K,L)
      GO TO 460
  760 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HMNAM,4HE   ,5,0,IW)
C
C     WRITE THE NEW NAME LIST TABLE ON THE DATA FILE
C
```

Figure 42. Subruotine MNAME Program Listing (Continued)

```
      IF(MFLAG.NE.3)CALL ERRM(4,4HMNAM,4HE    ,6,0,IW)
      CALL FILE(JN,INSERT,HEAD)
      WRITE(JN)NXN,NRN,NUN,
     1         (NNNS(I),(VNNS(I,J),J=1,2),
     2         (DESNS(I,J),J=1,10),(UNITNS(I,J),J=1,4),I=1,NXN),
     3         (NNNO(I),(VNNO(I,J),J=1,2),
     4         (DESNO(I,J),J=1,10),(UNITNO(I,J),J=1,4),I=1,NRN),
     5         (NNNI(I),(VNNI(I,J),J=1,2),
     6         (DESNI(I,J),J=1,10),(UNITNI(I,J),J=1,4),I=1,NUN)
      CALL FILE(JN,INSERT,MARK)
C
C     PRINT NAME LIST DATA
C
      IF(IPRINT.LT.2)RETURN
      CALL HPR(HEAD,IW)
      WRITE(IW,765)NXN,NRN,NUN
  765 FORMAT(//,1X,18HNUMBER OF STATES =,I2,//,1X,
     118HNUMBER OF OUTPUTS=,I2,//,1X,18HNUMBER OF INPUTS =,I2,//)
      WRITE(IW,770)
  770 FORMAT(//,20X,23H*** NAME LIST TABLE ***,/)
      WRITE(IW,780)
  780 FORMAT(/,1X,8HVARIABLE,6H NAME ,6X,13H DESCRIPTION ,
     131X,6H UNIT ,/)
C
C     PRINT NAME LIST DATA FOR STATES
C
      WRITE(IW,790)
  790 FORMAT(/,1X,6HSTATE ,/)
      WRITE(IW,800)(NNNS(I),(VNNS(I,J),J=1,2),(DESNS(I,J),J=1,10),
     1              (UNITNS(I,J),J=1,4),I=1,NXN)
  800 FORMAT(1X,I2,6X,2A4,4X,10A4,4X,4A4)
      IF(NDR.EQ.0)GO TO 940
C
C     PRINT NAME LIST DATA FOR DESIGN OUTPUTS
C
      WRITE(IW,810)
  810 FORMAT(/,1X,13HDESIGN OUTPUT,/)
      WRITE(IW,800)(NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
     1              (UNITNO(I,J),J=1,4),I=1,NDR)
  820 CONTINUE
      NDPR=NDR+NPR
      IF(NPR.EQ.0)GO TO 840
C
C     PRINT NAME LIST DATA FOR PERFORMANCE OUTPUTS
C
      WRITE(IW,830)
  830 FORMAT(/,1X,18HPERFORMANCE OUTPUT,/)
      NDRP1=NDR+1
      WRITE(IW,800)(NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
     1              (UNITNO(I,J),J=1,4),I=NDRP1,NDPR)
  840 CONTINUE
      IF(NSR.EQ.0)GO TO 860
C
C     PRINT NAME LIST DATA FOR SENSOR OUTPUTS
C
      WRITE(IW,850)
  850 FORMAT(/,1X,13HSENSOR OUTPUT,/)
      NDPRP1=NDPR+1
      WRITE(IW,800)(NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
     1              (UNITNO(I,J),J=1,4),I=NDPRP1,NRN)
  860 CONTINUE
      IF(NCL.EQ.0)GO TO 880
C
C     PRINT NAME LIST DATA FOR CONTROL INPUTS
C
      WRITE(IW,870)
```

Figure 42. Subroutine MNAME Program Listing (Continued)

```
  870 FORMAT(/,1X,13HCONTROL INPUT,/)
      WRITE(IW,800)(NNNI(I),(VNNI(I,J),J=1,2),(DESNI(I,J),J=1,10),
     1             (UNITNI(I,J),J=1,4),I=1,NCL)
  880 CONTINUE
      NCG=NCL+NGT
      IF(NGT.EQ.0)GO TO 900
C
C     PRINT NAME LIST DATA FOR GUST INPUTS
C
      WRITE(IW,890)
  890 FORMAT(/,1X,10HGUST INPUT,/)
      NCLP1=NCL+1
      WRITE(IW,800)(NNNI(I),(VNNI(I,J),J=1,2),(DESNI(I,J),J=1,10),
     1             (UNITNI(I,J),J=1,4),I=NCLP1,NCG)
  900 CONTINUE
      IF(NCO.EQ.0)GO TO 920
C
C     PRINT NAME LIST DATA FOR COMMAND INPUTS
C
      WRITE(IW,910)
  910 FORMAT(/,1X,13HCOMMAND INPUT,/)
      NCGP1=NCG+1
      WRITE(IW,800)(NNNI(I),(VNNI(I,J),J=1,2),(DESNI(I,J),J=1,10),
     1             (UNITNI(I,J),J=1,4),I=NCGP1,NUN)
  920 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(6,4HMNAM,4HE    ,5,0,IW)
      RETURN
  940 CONTINUE
C
C     PRINT NAME LIST DATA FOR OUTPUTS
C
      WRITE(IW,950)
  950 FORMAT(/,1X,6HOUTPUT,/)
      WRITE(IW,800)(NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
     1             (UNITNO(I,J),J=1,4),I=1,NRN)
C
C     PRINT NAME LIST DATA FOR INPUTS
C
      WRITE(IW,960)
  960 FORMAT(/,1X,5HINPUT,/)
      WRITE(IW,800)(NNNI(I),(VNNI(I,J),J=1,2),(DESNI(I,J),J=1,10),
     1             (UNITNI(I,J),J=1,4),I=1,NUN)
      IF(IPRINT.EQ.6)CALL DEBUG(7,4HMNAM,4HE    ,5,0,IW)
      RETURN
      END
```

Figure 42. Subroutine MNAME Program Listing (Concluded)

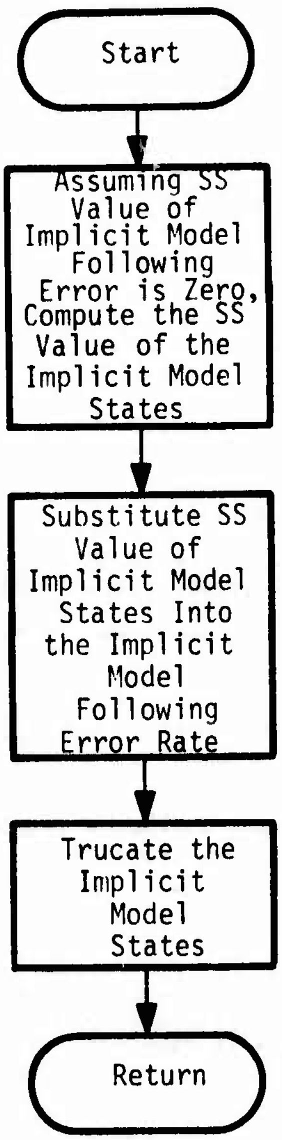

Figure 43. Subroutine IMRATE Flow Chart

```
      SUBROUTINE IMRATE(CM,DM,CW,DW,DUMMY1,DUMMY2,NX,NR,NU,
     1NXM,NRM,NUM,NXA,NRA,NUA,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
C
C     PURPOSE - TO OBTAIN IMPLICIT MODEL ERROR RATES
C     AND TRUNCATE THE IMPLICIT MODEL
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        MPRS
C        TDINVR
C
C     ARGUMENTS LIST
C        NX        INPUT     NUMBER OF STATES
C        NR        INPUT     NUMBER OF OUTPUTS
C        NU        INPUT     NUMBER OF INPUTS
C        NXA       INPUT     NO OF STATES WITHOUT IMPLICIT MODEL
C        NRA       INPUT     NO OF OUTPUTS WITHOUT IMPLICIT MODEL
C        NUA       INPUT     NO OF INPUTS WITHOUT IMPLICIT MODEL
C        IW        INPUT     FILE NO FOR LINE PRINTER
C        IPRINT    INPUT     PRINT CONTROL FLAG
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION CM(NRM,NXM),DM(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C
C     COMPUTE STEADY STATE VALUE OF IMPLICIT MODEL STATES
C     FOR ZERO MODEL ERROR
C
      NXR=NX-NXA
      DO 160 I=1,NXR
      DO 120 J=1,NXR
      JJ=NXA+J
  120 DUMMY1(I,J)=CW(I,JJ)
      DO 140 J=1,NXA
      JJ=NXR+J
  140 DUMMY1(I,JJ)=CW(I,J)
      DO 160 J=1,NU
      JJ=NX+J
  160 DUMMY1(I,JJ)=DW(I,J)
      NDR=NXR
      NDC=NXR+NXA+NU
      IF(IPRINT.LT.6)GO TO 170
      CALL MPRS(CW,NRM,NXM,NXR,NX,0,0,4HCW   )
      CALL MPRS(DW,NRM,NUM,NXR,NU,0,0,4HDW   )
      CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,0,0,4HDMY1)
  170 CONTINUE
      CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)
      IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 240
      WRITE(IW,180)ISOL,IDSOL
  180 FORMAT(1H1,//,1X,14HTDINVR FAILURE,6H ISOL=,I2,7H IDSOL=,I2)
      WRITE(IW,200)
  200 FORMAT(//,1X,43HSTEADY STATE VALUE OF MODEL STATE CANNOT BE,
     11X,23HCOMPUTED FOR ZERO ERROR)
      WRITE(IW,220)
  220 FORMAT(//,1X,26HMODEL STATES ARE TRUNCATED)
C
C     COMPUTE IMPLICIT MODEL ERROR RATES
C
  240 CONTINUE
      DO 260 I=1,NR
      DO 260 J=1,NX
  260 DUMMY2(I,J)=CM(I,J)
```

Figure 44. Subroutine IMRATE Program Listing

```
      DO 280 I=1,NR                                                     IMRATE65
      DO 280 J=1,NXA                                                    IMRATE66
      JJ=NXR+J                                                          IMRATE67
      DO 280 K=1,NXR                                                    IMRATE68
      KK=NXA+K                                                          IMRATE69
  280 CM(I,J)=CM(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)                         IMRATE70
      DO 300 I=1,NR                                                     IMRATE71
      DO 300 J=1,NU                                                     IMRATE72
      JJ=NX+J                                                           IMRATE73
      DO 300 K=1,NXR                                                    IMRATE74
      KK=NXA+K                                                          IMRATE75
  300 DM(I,J)=DM(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)                         IMRATE76
      NX=NXA                                                            IMRATE77
      IF(IPRINT.LT.6)GO TO 320                                          IMRATE78
      CALL MPRS(CM,NRM,NXM,NR,NXA,0.0,4HCM  )                           IMRATE79
      CALL MPRS(DM,NRM,NUM,NR,NU,0.0,4HDM  )                            IMRATE80
      CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,0.0,4HDMY1)                  IMRATE81
  320 CONTINUE                                                          IMRATE82
      RETURN                                                            IMRATE83
      END                                                               IMRATE84
```

Figure 44. Subroutine IMRATE Program Listing (Concluded)

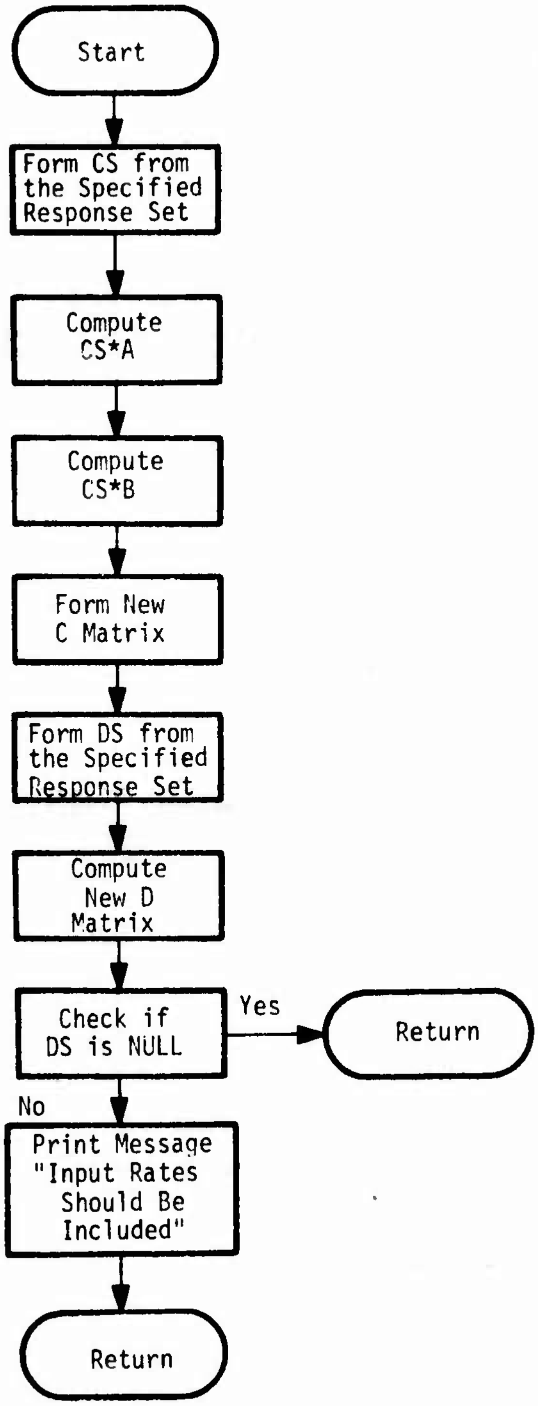

Figure 45. Subroutine DIFFK Flow Chart

```
      SUBROUTINE DIFFK(A,B,C,D,DUMMY1,DUMMY2,NX,NR,NU,
     1NXM,NRM,NUM,NRS,IRS,ID,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
C
C     PURPOSE - TO OBTAIN DERIVATIVES OF RESPONSES
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        NX          INPUT      NUMBER OF STATES
C        NR          INPUT      NUMBER OF OUTPUTS
C        NU          INPUT      NUMBER OF INPUTS
C        NRS         INPUT      NO OF RESPONSES TO BE DIFFERENTIATED
C        ID          INPUT      CONTROLS ENTRY POINT IN THE SUBROUTINE
C        IW                     FILE NO FOR LINE PRINTER
C        IPRINT                 PRINT CONTROL FLAG
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION IRS(NRM),DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
      NUS=NU
      IF(ID.GT.1)GO TO 140
C
C     OBTAIN FIRST DERIVATIVES ONLY
C
C     FORM CS MATRIX
C
      DO 10 I=1,NRS
      II=IRS(I)
      DO 10 J=1,NX
   10 DUMMY1(I,J)=C(II,J)
C
C     COMPUTE CS*A MATRIX
C
      DO 30 I=1,NRS
      DO 30 J=1,NX
      DUMMY2(I,J)=0.0
      DO 30 K=1,NX
   30 DUMMY2(I,J)=DUMMY2(I,J)+DUMMY1(I,K)*A(K,J)
C
C     COMPUTE NEW C MATRIX
C
      DO 50 I=1,NRS
      II=NR+I
      DO 50 J=1,NX
   50 C(II,J)=DUMMY2(I,J)
C
C     FORM CS*B MATRIX
C
      DO 60 I=1,NRS
      DO 60 J=1,NU
      DUMMY2(I,J)=0.0
      DO 60 K=1,NX
   60 DUMMY2(I,J)=DUMMY2(I,J)+DUMMY1(I,K)*B(K,J)
C
C     FORM DS MATRIX
C
      DO 70 I=1,NRS
      II=IRS(I)
      DO 70 J=1,NU
   70 DUMMY1(I,J)=D(II,J)
C
C     COMPUTE NEW D MATRIX
C
```

Figure 46. Subroutine DIFFK Program Listing

```
      DO 80 I=1,NRS
      DO 80 J=1,NU
      II=NR+I
   80 D(II,J)=DUMMY2(I,J)
C
C     CHECK IF DS MATRIX IS NULL
C
      DO 90 I=1,NRS
      DO 90 J=1,NRS
      IF(DUMMY1(I,J).NE.0.0)GO TO 100
   90 CONTINUE
      NR=NR+NRS
      RETURN
C
C     PRINT A MESSAGE THAT THE INPUT RATES ARE NECESSARY FOR
C     CORRECTLY OBTAINING THE DERIVATIVES OF THE RESPONSES
C
  100 CONTINUE
      WRITE(IW,120)
  120 FORMAT(1H1,//,1X,85H*** THE INPUT RATES SHOULD BE INCLUDED IN TAKI
     1NG THE DERIVATIVES OF THE RESPONSES ***,//)
      RETURN
  140 CONTINUE
C
C     OBTAIN FIRST AND SECOND DERIVATIVES
C
      WRITE(IW,160)
  160 FORMAT(//,1X,55H*** THE SECOND DERIVATIVE OPTION IS NOT IMPLEMENTE
     1D ***,//)
      RETURN
      END
```

Figure 46. Subroutine DIFFK Program Listing (Concluded)

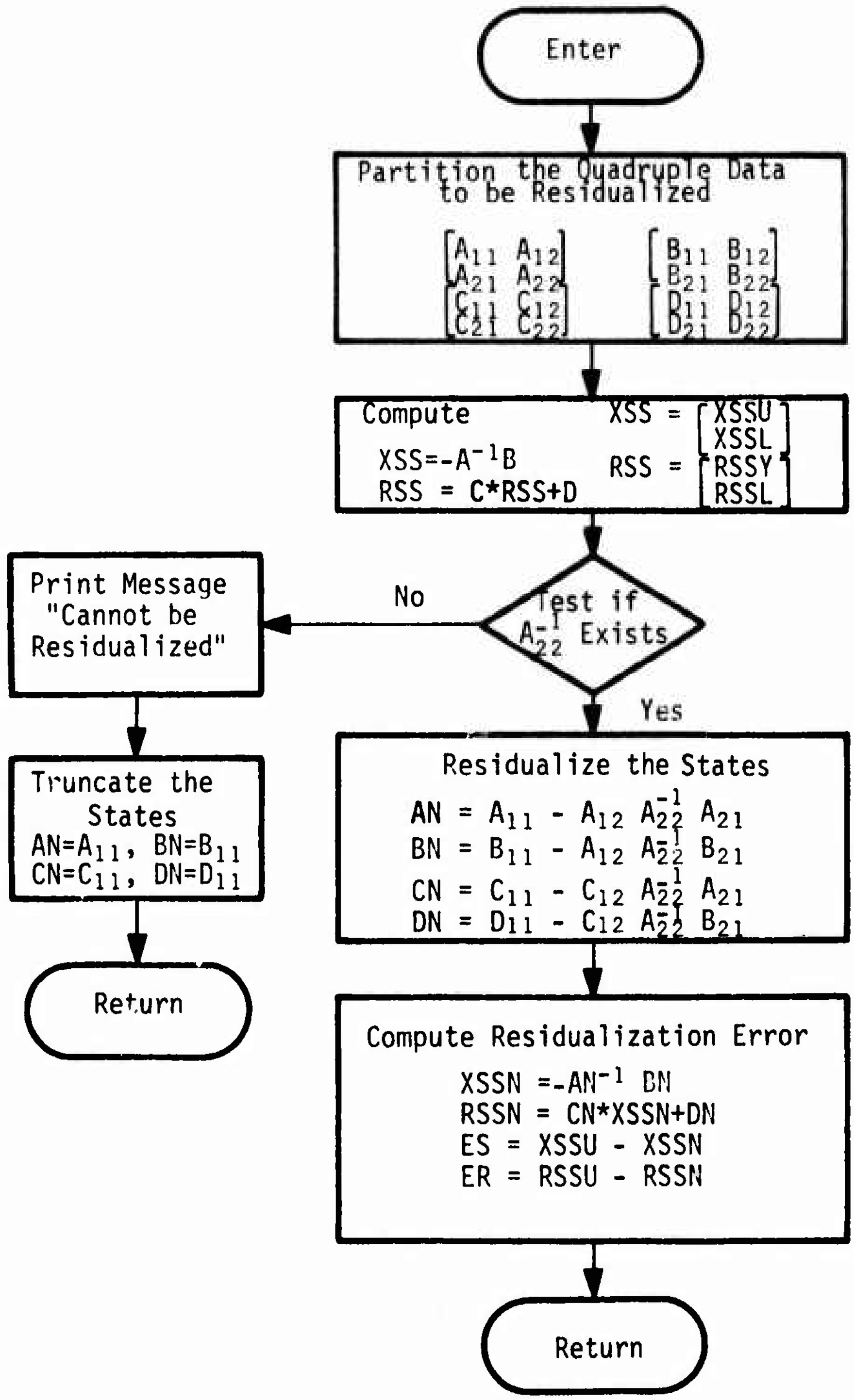

Figure 47. Subroutine REDUCE Flow Chart

```
      SUBROUTINE REDUCE(A,B,C,D,DUMMY1,DUMMY2,ES,ER,
     1NX,NR,NU,NXR,NRR,NRT,NXM,NRM,NUM,T,IW,IPRINT,
     2NDM11,NDM12,NDM21,NDM22)
C
C     PURPOSE - TO REDUCE THE ORDER OF STATE SPACE DATA
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        TDINVR
C        MPRS
C
C     ARGUMENT LIST
C        NX          INPUT       NUMBER OF STATES
C        NR          INPUT       NUMBER OF OUTPUTS
C        NU          INPUT       NUMBER OF INPUTS
C        NXR         INPUT       NO OF STATES TO BE RESIDUALIZED
C        NRR         INPUT       NO OF OUTPUTS TO BE RESIDUALIZED
C        NRT         INPUT       NO OF OUTPUTS TO BE TRUNCATED
C        T           INPUT       SAMPLE TIME
C        IW          INPUT       FILE NO FOR LINE PRINTER
C        IPRINT      INPUT       PRINT CONTROL FLAG
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
      DIMENSION ES(NXM,NUM),ER(NRM,NUM)
      NXN=NX-NXR
      NRN=NRT+NRR
      NUN=NU
C
C     COMPUTE SS VALUE OF STATE AND OUTPUT FOR ORIGINAL SYSTEM
C
      IF(IPRINT.LT.5)GO TO 260
      DO 140 I=1,NX
      DO 140 J=1,NX
  140 DUMMY1(I,J)=A(I,J)
      DO 160 I=1,NX
      DO 160 J=1,NU
      JJ=NX+J
  160 DUMMY1(I,JJ)=B(I,J)
      NDR=NX
      NDC=NX+NU
      IF(IPRINT.GE.6)CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,T,4HDMY1)
      CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)
      IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 200
      WRITE(IW,180)ISOL,IDSOL
  180 FORMAT(/,1X,14HTDINVR FAILURE,6H ISOL=,I2,7H IDSOL=,I2)
      WRITE(IW,190)
  190 FORMAT(/,1X,40HRESIDUALIZATION ERROR CANNOT BE COMPUTED,
     1/,1X,30HSINCE SS VALUES DOES NOT EXIST)
      JPRINT=2
      GO TO 260
  200 CONTINUE
      JPRINT=IPRINT
      DO 220 I=1,NX
      DO 220 J=1,NU
      JJ=NX+J
  220 ES(I,J)=-DUMMY1(I,JJ)
      DO 240 I=1,NR
      DO 240 J=1,NU
      ER(I,J)=D(I,J)
      DO 240 K=1,NX
```

Figure 48. Subroutine REDUCE Program Listing

```
  240 ER(I,J)=ER(I,J)+C(I,K)*ES(K,J)
      IF(IPRINT.GE.6)CALL MPRS(ES,NXM,NUM,NX,NU,T,4HES  )
      IF(IPRINT.GE.6)CALL MPRS(ER,NRM,NUM,NR,NU,T,4HER  )
  260 CONTINUE
C
C     QUADRUPLE DATA IS REDUCED BY RESIDUALIZATION
C
      DO 280 I=1,NXR
      II=NXN+I
      DO 280 J=1,NXR
      JJ=NXN+J
  280 DUMMY1(I,J)=A(II,JJ)
      DO 300 I=1,NXR
      II=NXN+I
      DO 300 J=1,NXN
      JJ=NXR+J
  300 DUMMY1(I,JJ)=A(II,J)
      DO 320 I=1,NXR
      II=NXN+I
      DO 320 J=1,NUN
      JJ=NXR+NXN+J
  320 DUMMY1(I,JJ)=B(II,J)
      NDR=NXR
      NDC=NXR+NXN+NUN
      IF(IPRINT.GE.6)CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,T,4HDMY1)
      CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)
      IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 360
C
C     RESIDUALIZATION IS NOT POSSIBLE AND SO
C     QUADRUPLE DATA IS REDUCED BY TRUNCATION
C
      WRITE(IW,189)ISOL,IDSOL
      WRITE(IW,340)
  340 FORMAT(/,1X,47HCANNOT BE RESIDUALIZED SINCE SS VALUE OF STATES,
     1/,1X,31HBEING ELIMINATED DOES NOT EXIST)
      WRITE(IW,350)
  350 FORMAT(/,1X,31HQ DATA IS REDUCED BY TRUNCATION)
      NX=NXN
      NR=NRN
      NU=NUN
      RETURN
C
C     COMPUTE RESIDUALIZED QUADRUPLES
C
  360 CONTINUE
      DO 380 I=1,NX
      DO 380 J=1,NX
  380 DUMMY2(I,J)=A(I,J)
      DO 400 I=1,NXN
      DO 400 J=1,NXN
      JJ=NXR+J
      DO 400 K=1,NXR
      KK=NXN+K
  400 A(I,J)=A(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)
      DO 420 I=1,NXN
      DO 420 J=1,NUN
      JJ=NXR+NXN+J
      DO 420 K=1,NXR
      KK=NXN+K
  420 B(I,J)=B(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)
      DO 440 I=1,NR
      DO 440 J=1,NX
  440 DUMMY2(I,J)=C(I,J)
      DO 460 I=1,NRR
      DO 460 J=1,NXN
      JJ=NXR+J
```

Figure 48. Subroutine REDUCE Program Listing (Continued)

```
      DO 460 K=1,NXR
      KK=NXN+K
  460 C(I,J)=C(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)
      DO 480 I=1,NRR
      DO 480 J=1,NUN
      JJ=NXR+NXN+J
      DO 480 K=1,NXR
      KK=NXN+K
  480 D(I,J)=D(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)
C
C     COMPUTE SS VALUE OF STATE AND OUTPUT FOR REDUCED SYSTEM
C     AND SUBTRACT IT FROM SS VALUE OBTAINED EARLIER TO GET
C     THE ERROR OF RESIDUALIZATION
C
      IF(JPRINT.LT.3)GO TO 600
      DO 500 I=1,NXN
      DO 500 J=1,NXN
  500 DUMMY1(I,J)=A(I,J)
      DO 520 I=1,NXN
      DO 520 J=1,NUN
      JJ=NXN+J
  520 DUMMY1(I,JJ)=B(I,J)
      NDR=NXN
      NDC=NXN+NUN
      IF(IPRINT.GE.6)CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,T,4HDMY1)
      CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)
      IF((ISOL.GT.1).OR.(IDSOL.GT.1))GO TO 620
C
C     COMPUTE RESIDUALIZATION ERROR
C
      DO 540 I=1,NXN
      DO 540 J=1,NUN
      JJ=NXN+J
  540 ES(I,J)=ES(I,J)+DUMMY1(I,JJ)
      DO 560 I=1,NRN
      DO 560 J=1,NUN
      ER(I,J)=-D(I,J)+ER(I,J)
      JJ=NXN+J
      DO 560 K=1,NXN
  560 ER(I,J)=ER(I,J)+C(I,K)*DUMMY1(K,JJ)
      WRITE(IW,580)
  580 FORMAT(/,1X,40HRESIDUALIZATION ERROR MATRICES ES AND ER,/)
      CALL MPRS(ES,NXM,NUM,NXN,NUN,T,4HES  )
      CALL MPRS(ER,NRM,NUM,NRN,NUN,T,4HER  )
  600 CONTINUE
      NX=NXN
      NR=NRN
      NU=NUN
      RETURN
  620 CONTINUE
      STOP 0030
      END
```

Figure 48. Subroutine REDUCE Program Listing (Concluded)

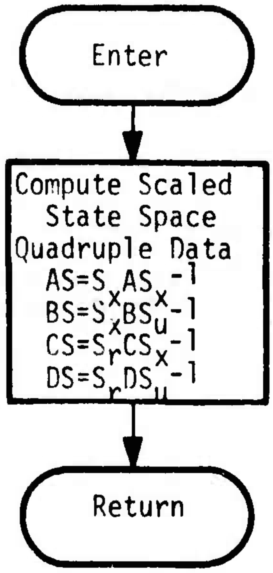

Figure 49. Subroutine SCAL Flow Chart

```
      SUBROUTINE SCAL(A,B,C,D,SCFS,SCFO,SCFI,
     1NXN,NRN,NUN,NXM,NRM,NUM)
C
C     PURPOSE - TO COMPUTE SCALED QUADRUPLES
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENT LIST
C        SCFS       INPUT     SCALING ARRAY FOR STATE
C        SCFO       INPUT     SCALING ARRAY FOR OUTPUT
C        SCFI       INPUT     SCALING ARRAY FOR INPUT
C        NXN        INPUT     NUMBER OF REDUCED STATES
C        NRN        INPUT     NUMBER OF REDUCED OUTPUTS
C        NUN        INPUT     NUMBER OF REDUCED INPUTS
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION SCFS(NXM),SCFO(NRM),SCFI(NUM)
      DO 15  I=1,NXN
      DO 15  J=1,NXN
  150 A(I,J)=SCFS(I)*A(I,J)/SCFS(J)
      DO 17  I=1,NXN
      DO 17  J=1,NUN
  170 B(I,J)=SCFS(I)*B(I,J)/SCFI(J)
      DO 19  I=1,NRN
      DO 19  J=1,NXN
  190 C(I,J)=SCFO(I)*C(I,J)/SCFS(J)
      DO 21  I=1,NRN
      DO 21  J=1,NUN
  210 D(I,J)=SCFO(I)*D(I,J)/SCFI(J)
      RETURN
      END
```

Figure 50. Subroutine SCAL Program Listing

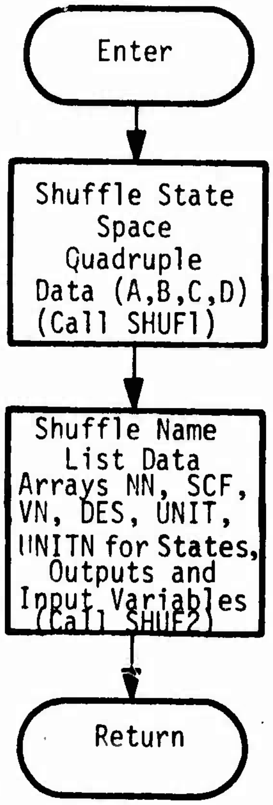

Figure 51. Subroutine SHUFF Flow Chart

```
      SUBROUTINE SHUFF(A,B,C,D,NNS,VNS,DESS,UNITS,
     1NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,
     2NSHUFS,NSHUFO,NSHUFI,DUMMY1,DUMMY2,NX,NR,NU,NXM,NRM,NUM,
     3NDM11,NDM12,NDM21,NDM22)
C
C     PURPOSE - TO SHUFFLE QUADRUPLE DATA AND NAME LIST DATA
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - MAY, 1975
C
C     SUBPROGRAMS CALLED
C        SHUF1
C        SHUF2
C
C     ARGUMENT LIST
C        NX          INPUT      NUMBER OF STATES
C        NR          INPUT      NUMBER OF OUTPUTS
C        NU          INPUT      NUMBER OF INPUTS
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
      DIMENSION NSHUFS(NXM),NSHUFO(NRM),NSHUFI(NUM)
      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C
C     SHUFFLE A B C D MATRICES
C
      CALL SHUF1(A,NSHUFS,NSHUFS,DUMMY1,NXM,NXM,NX,NX,NDM11,NDM12)
      CALL SHUF1(B,NSHUFS,NSHUFI,DUMMY1,NXM,NUM,NX,NU,NDM11,NDM12)
      CALL SHUF1(C,NSHUFO,NSHUFS,DUMMY1,NRM,NXM,NR,NX,NDM11,NDM12)
      CALL SHUF1(D,NSHUFO,NSHUFI,DUMMY1,NRM,NUM,NR,NU,NDM11,NDM12)
C
C     SHUFFLE SCALING, UNIT AND DESCRIPTION ARRAYS
C
      CALL SHUF2(NNS,VNS,DESS,UNITS,NSHUFS,DUMMY1,DUMMY2,
     1NXM,NX,NDM11,NDM12,NDM21,NDM22)
      CALL SHUF2(NNO,VNO,DESO,UNITO,NSHUFO,DUMMY1,DUMMY2,
     1NRM,NR,NDM11,NDM12,NDM21,NDM22)
      CALL SHUF2(NNI,VNI,DESI,UNITI,NSHUFI,DUMMY1,DUMMY2,
     1NUM,NU,NDM11,NDM12,NDM21,NDM22)
      RETURN
      END
```

Figure 52. Subroutine SHUFF Program Listing

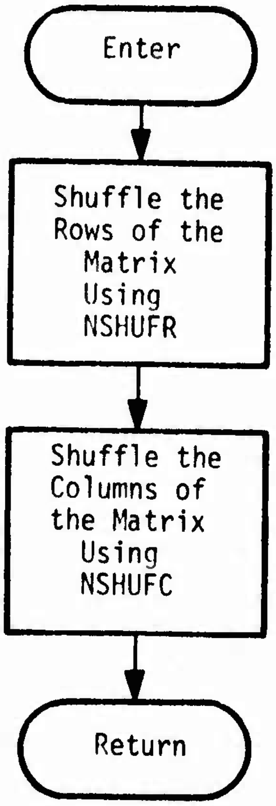

Figure 53. Subroutine SHUFF1 Flow Chart

```
      SUBROUTINE SHUF1(ABCD,NSHUFR,NSHUFC,DUMMY1,NRM,NCM,NR,NC,
     1NDM11,NDM12)
C
C     PURPOSE - TO SHUFFLE THE MATRIX ABCD
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENT LIST
C        ABCD       IN/OUT     MATRIX TO BE SHUFFLED
C        NSHUFR     INPUT      ROW SHUFFLING ARRAY
C        NSHUFC     INPUT      COLUMN SHUFFLING ARRAY
C        NRM        INPUT      MAXIMUM NUMBER OF ROWS
C        NCM        INPUT      MAXIMUM NUMBER OF COLUMNS
C        NR         INPUT      NUMBER OF ROWS
C        NC         INPUT      NUMBER OF COLUMNS
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION ABCD(NRM,NCM),DUMMY1(NDM11,NDM12),NSHUFR(NRM)
      DIMENSION NSHUFC(NCM)
      DO 120 I=1,NR
      II=NSHUFR(I)
      DO 120 J=1,NC
  120 DUMMY1(I,J)=ABCD(II,J)
      DO 140 J=1,NC
      JJ=NSHUFC(J)
      DO 140 I=1,NR
  140 ABCD(I,J)=DUMMY1(I,JJ)
      RETURN
      END
```

Figure 54. Subroutine SHUF1 Program Listing

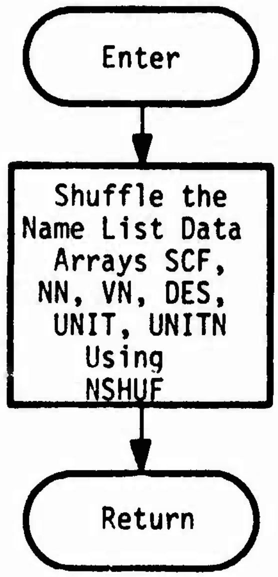

Figure 55. Subroutine SHUF2 Flow Chart

```
      SUBROUTINE SHUF2(NN,VN,DES,UNIT,NSHUF,DUMMY3,DUMMY1,
     1NM,N,NDM11,NDM12,NDM21,NDM22)
C
C     PURPOSE - TO SHUFFLE NAME LIST ARRAYS
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENT LIST
C        NN         IN/OUT     NUMBER ARRAY
C        VN         IN/OUT     VARIABLE NAME ARRAY
C        DES        IN/OUT     DESCRIPTION ARRAY
C        UNIT       IN/OUT     UNIT ARRAY
C        NSHUF      INPUT      SHUFFLING ARRAY
C        NM         INPUT      MAXIMUM NUMBER OF SYSTEM VARIABLES
C        N          INPUT      NUMBER OF SYSTEM VARIABLES
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION NN(NM),VN(NM,2),DES(NM,10),UNIT(NM,4)
      DIMENSION DUMMY3(NDM11,NDM12),DUMMY1(NDM21,NDM22)
      DIMENSION NSHUF(NM)
      INTEGER DUMMY1
      DO 16  I=1,N
      II=NSHUF(I)
      DUMMY1(1,I)=NN(II)
      DO 12  J=1,2
      JJ=1+J
  120 DUMMY3(JJ,I)=VN(II,J)
      DO 140 J=1,10
      JJ=3+J
  140 DUMMY3(JJ,I)=DES(II,J)
      DO 160 J=1,4
      JJ=13+J
  160 DUMMY3(JJ,I)=UNIT(II,J)
      DO 240 I=1,N
      NN(I)=DUMMY1(1,I)
      DO 200 J=1,2
      JJ=1+J
  200 VN(I,J)=DUMMY3(JJ,I)
      DO 220 J=1,10
      JJ=3+J
  220 DES(I,J)=DUMMY3(JJ,I)
      DO 240 J=1,4
      JJ=13+J
  240 UNIT(I,J)=DUMMY3(JJ,I)
      RETURN
      END
```

Figure 56. Subroutine SHUF2 Program Listing

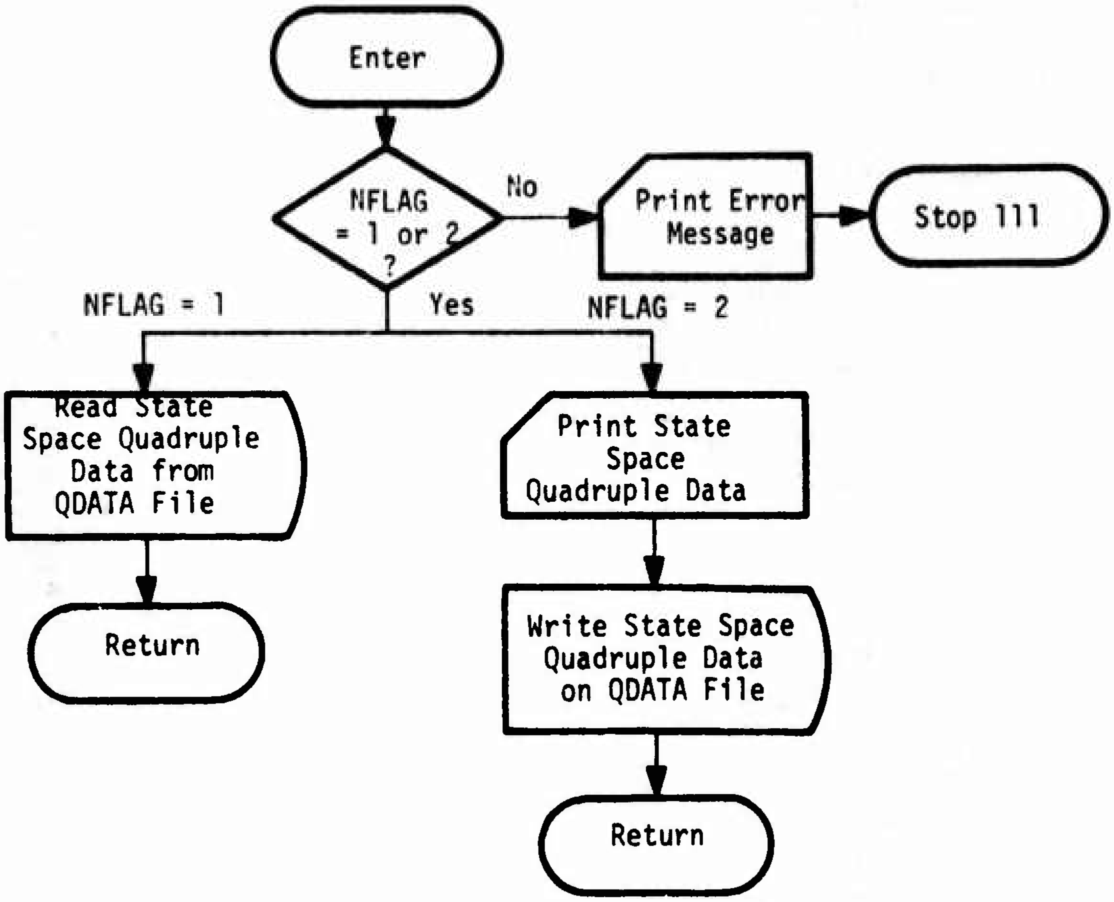

Figure 57. Subroutine QDIO Flow Chart

```
      SUBROUTINE QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
     1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,LABEL,MARK,
     2LOCATE,NULL,INSERT,NFLAG)
C
C     PURPOSE - TO READ AND WRITE QUADRUPLE DATA
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        ERRM
C        MPRS
C        FILE
C
C     ARGUMENTS LIST
C        A          IN/OUT     STATE TRANSITION MATRIX
C        B          IN/OUT     CONTROL INPUT MATRIX
C        C          IN/OUT     STATE OUTPUT MATRIX
C        D          IN/OUT     CONTROL OUTPUT MATRIX
C        Q          IN/OUT     QUADRATIC WEIGHTS MATRIX
C        NX         IN/OUT     NUMBER OF STATES
C        NR         IN/OUT     NUMBER OF OUTPUTS
C        NU         IN/OUT     NUMBER OF INPUTS
C        NXM        INPUT      MAXIMUM NUMBER OF STATES
C        NRM        INPUT      MAXIMUM NUMBER OF OUTPUTS
C        NUM        INPUT      MAXIMUM NUMBER OF INPUTS
C        NXA        IN/OUT     NO OF STATES WITHOUT IMPLICIT MODEL
C        NRA        IN/OUT     NO OF OUTPUTS WITHOUT IMPLICIT MODEL
C        NUA        IN/OUT     NO OF INPUTS WITHOUT IMPLICIT MODEL
C        NR1        IN/OUT     NO OF DESIGN OUTPUTS
C        NR2        IN/OUT     NO OF PERFORMANCE OUTPUTS
C        NR3        IN/OUT     NO OF SENSOR OUTPUTS
C        NU1        IN/OUT     NO OF CONTROL INPUTS
C        NU2        IN/OUT     NO OF GUST INPUTS
C        NU3        IN/OUT     NO OF COMMAND INPUTS
C        T          IN/OUT     SAMPLE TIME
C        IQ         INPUT      FLAG INDICATING USAGE OF Q
C        IPRINT     INPUT      PRINT CONTROL FLAG
C        IW         INPUT      FILE NO FOR LINE PRINTER
C        JQ         INPUT      FILE NO FOR QUADRUPLE DATA FILE
C        LABEL      INPUT      LABEL NAME FOR QUADRUPLE DATA
C        MARK       INPUT      HOLLERITH $$..$
C        LOCATE     INPUT      HOLLERITH LOCA
C        NULL       INPUT      HOLLERITH NULL
C        INSERT     INPUT      HOLLERITH INSE
C        NFLAG      INPUT      CONTROLS ENTRY POINT IN THE SUBROUTINE
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION Q(NRM,NRM),LABEL(20),MARK(20)
      IF(NFLAG.NE.2)GO TO 220
      IF(IPRINT.LT.2)GO TO 200
C
C     PRINT QUADRUPLE DATA
C
      WRITE(IW,120)
  120 FORMAT(//,20X,22H*** QUADRUPLE DATA ***,/)
      IF(T.NE.0.0)GO TO 140
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA    )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HB    )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC    )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD    )
      GO TO 160
  140 CONTINUE
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HF    )
```

Figure 58. Subroutine QDIO Program Listing

```
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HG    )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HH    )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HE    )
  160 CONTINUE
      IF(IQ.NE.1)GO TO 200
C
C     PRINT WEIGHTING MATRIX Q
C
      WRITE(IW,180)
  180 FORMAT(//,20X,47H*** STARTING WEIGHTS FOR OPTIMAL CONTROL DESIGN,
     14H ***,/)
      CALL MPRS(Q,NRM,NRM,NR,NR,T,4HQ0   )
  200 CONTINUE
      CALL FILE(JQ,INSERT,LABEL)
      IF(IQ.NE.1)GO TO 210
C
C     WRITE QUADRUPLE DATA AND WEIGHTING MATRIX Q ON FILE QDATA
C
      WRITE(JQ)T,NX,NR,NU,
     1((A(I,J),I=1,NX),J=1,NX),
     2((B(I,J),I=1,NX),J=1,NU),
     3((C(I,J),I=1,NR),J=1,NX),
     4((D(I,J),I=1,NR),J=1,NU),
     5NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3,
     6((Q(I,J),I=1,NR1),J=1,NR1)
      CALL FILE(JQ,INSERT,MARK)
      RETURN
  210 CONTINUE
C
C     WRITE QUADRUPLE DATA ON FILE QDATA
C
      WRITE(JQ)T,NX,NR,NU,
     1((A(I,J),I=1,NX),J=1,NX),
     2((B(I,J),I=1,NX),J=1,NU),
     3((C(I,J),I=1,NR),J=1,NX),
     4((D(I,J),I=1,NR),J=1,NU),
     5NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3
      CALL FILE(JQ,INSERT,MARK)
      RETURN
  220 CONTINUE
      IF(NFLAG.NE.1)CALL ERRM(1,4HQDIO,4H    ,0,0,IW)
      IF(IPRINT.EQ.6)WRITE(IW,225)LABEL
  225 FORMAT(1X,20A4)
      CALL FILE(JQ,LOCATE,LABEL)
      IF(IQ.NE.1)GO TO 230
C
C     READ QUADRUPLE DATA AND WEIGHTING MATRIX Q FROM FILE QDATA
C
      READ(JQ)T,NX,NR,NU,
     1((A(I,J),I=1,NX),J=1,NX),
     2((B(I,J),I=1,NX),J=1,NU),
     3((C(I,J),I=1,NR),J=1,NX),
     4((D(I,J),I=1,NR),J=1,NU),
     5NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3,
     6((Q(I,J),I=1,NR1),J=1,NR1)
      RETURN
C
C     READ QUADRUPLE DATA FROM FILE QDATA
C
  230 CONTINUE
      READ(JQ)T,NX,NR,NU,
     1((A(I,J),I=1,NX),J=1,NX),
     2((B(I,J),I=1,NX),J=1,NU),
     3((C(I,J),I=1,NR),J=1,NX),
     4((D(I,J),I=1,NR),J=1,NU),
     5NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3
```

Figure 58. Subroutine QDIO Program Listing (Continued)

```
      RETURN                                                            QDIO 131
      END                                                               QDIO 132
```

Figure 58. Subroutine QDIO Program Listing (Concluded)

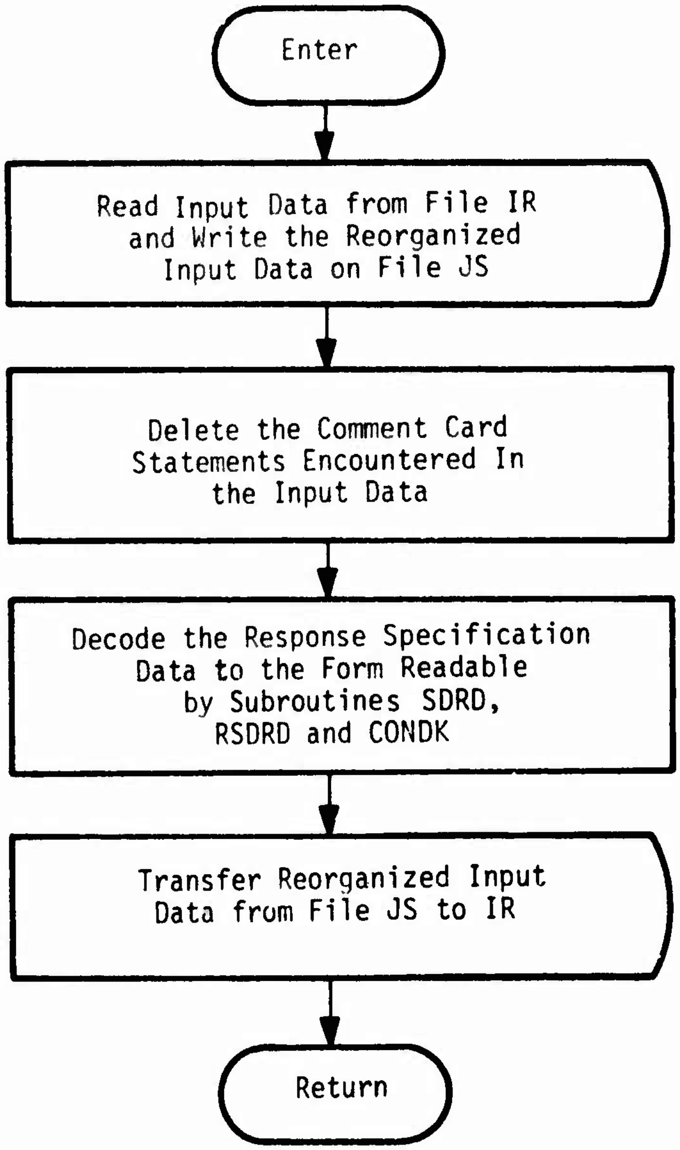

Figure 59. Subroutine IDRO Flow Chart

```
      SUBROUTINE IDRO(IR,IW,JS)
C
C     PURPOSE - TO REORGANIZE INPUT DATA
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        IR          INPUT      FILE NO FOR CARD READER BUFFER
C        IW          INPUT      FILE NO FOR LINE PRINTER
C        JS          INPUT      FILE NO FOR SCRATCH FILE
C
      DIMENSION CARD1(20),CARD2(80),CARD3(76),CARD4(20),CHEAD(6)
      DATA CHEAD /4HCONS,4HSELE,4HRETA,4HTRUN,4HREST,4HSCAL/
      DATA HBBBB,HENDB,HB,HBB,HEB /4H    ,4HEND ,1H ,1H(,1H)/
      DATA HCOM,HDAS,HDOT /1H,,1H-,1H./
      DATA HX,HR,HU/1HX,1HR,1HU/
      DATA HCB,HE/2HC ,1HE/
      NID=6
      REWIND IR
      REWIND JS
      CARD3(2)=HBB
      CARD3(4)=HEB
      DO 100 I=5,76
  100 CARD3(I)=HB
      CARD4(1)=HENDB
      DO 110 I=2,20
  110 CARD4(I)=HBBBB
C
C     READ CARD IMAGES FROM FILE IR AND IGNORE THE COMMENT CARDS
C
  120 CONTINUE
      READ(IR,140)CARD1
  140 FORMAT(20A4)
      IF(EOF(IR))540,160
  160 CONTINUE
      DECODE(4,170,CARD1(1))CC,DUMMY
  170 FORMAT(A2,A2)
      IF(CC.EQ.HCB)GO TO 120
      WRITE(JS,140)CARD1
      DO 180 J=1,NID
      IF(CARD1(1).EQ.CHEAD(J))GO TO 200
  180 CONTINUE
      GO TO 120
C
C     READ RESPONSE SPECIFICATION DATA AND ENCODE
C     INTO SIMPLER RESPONSE SPECIFICATIONS
C
  200 CONTINUE
      READ(IR,220)CARD2
  220 FORMAT(80A1)
      I=0
  240 CONTINUE
      I=I+1
      IF(I.GE.81)GO TO 200
      IF(CARD2(I).EQ.HE)GO TO 520
      IF(CARD2(I).EQ.HB)GO TO 240
      IF(CARD2(I).EQ.HCOM)GO TO 240
      IF(CARD2(I).EQ.HDAS)GO TO 400
      IF(CARD2(I).EQ.HDOT)GO TO 520
      IF((CARD2(I).NE.HX).AND.(CARD2(I).NE.HR).AND.(CARD2(I).NE.HU))
     1GO TO 240
      I=I+1
      IF(CARD2(I).EQ.HBB)GO TO 280
```

Figure 60. Subroutine IDRO Program Listing

```
      I=I+1
      IF(CARD2(I).EQ.HBP)GO TO 26
      I=I+2
      IF(CARD2(I).NE.HBP)GO TO 62
      ENCODE(4,250,CARD3(1))CARD2(I-4),CARD2(I-3),CARD2(I-2),CARD2(I-1)
  250 FORMAT(4A1)
      GO TO 300
  260 CONTINUE
      ENCODE(4,250,CARD3(1))CARD2(I-2),CARD2(I-1),HB,HB
      GO TO 300
  280 CONTINUE
      ENCODE(4,250,CARD3(1))CARD2(I-1),HB,HB,HB
  300 CONTINUE
      I=I+2
      IF(CARD2(I).EQ.HEP)GO TO 32
      I=I+1
      IF(CARD2(I).NE.HEP)GO TO 62
      ENCODE(2,310,CARD3(3))CARD2(I-2),CARD2(I-1)
  310 FORMAT(2A1)
      GO TO 340
  320 CONTINUE
      ENCODE(2,310,CARD3(3))HB,CARD2(I-1)
  340 CONTINUE
      DECODE(2,360,CARD3(3))NP
  360 FORMAT(I2)
      IF(CARD1(1).EQ.CHEAD(6))GO TO 380
      WRITE(JS,370)CARD3
  370 FORMAT(A4,A1,A2,73A1)
      GO TO 240
  380 CONTINUE
      WRITE(JS,390)(CARD3(I),I=1,4),(CARD2(I),I=9,80)
  390 FORMAT(A4,A1,A2,73A1)
      GO TO 200
  400 CONTINUE
      I=I+2
      IF(CARD2(I).NE.HBP)I=I+1
      IF(CARD2(I).NE.HBP)I=I+2
      IF(CARD2(I).NE.HBP)GO TO 62
      I=I+2
      IF(CARD2(I).EQ.HEP)GO TO 44
      I=I+1
      IF(CARD2(I).NE.HEP)GO TO 62
      ENCODE(2,310,CARD3(3))CARD2(I-2),CARD2(I-1)
      GO TO 460
  440 CONTINUE
      ENCODE(2,310,CARD3(3))HB,CARD2(I-1)
  460 CONTINUE
      DECODE(2,360,CARD3(3))NN
      IF(NN.LE.NP)GO TO 520
      NPP1=NP+1
      DO 500 J=NPP1,NN
      ENCODE(2,480,CARD3(3))J
  480 FORMAT(I2)
      WRITE(JS,370)CARD3
  500 CONTINUE
      GO TO 240
  520 CONTINUE
      WRITE(JS,140)CARD4
      GO TO 120
  540 CONTINUE
      ENDFILE JS
      REWIND IR
      REWIND JS
C
C     TRANSFER THE CARD IMAGES FROM FILE JS TO FILE IR
C
```

Figure 60. Subroutine IDRO Program Listing (Continued)

```
  560 CONTINUE
      READ(JS,140)CARD1
      IF(EOF(JS))600,580
  580 CONTINUE
      WRITE(IR,140)CARD1
      GO TO 560
  600 CONTINUE
      ENDFILE IR
      REWIND IR
      REWIND JS
      RETURN
C
C     PRINT ERROR MESSEGE
C
  620 CONTINUE
      WRITE(IW,640)CARD2
  640 FORMAT(1H1,//,1X,32HERROR IN REORGANIZING INPUT DATA,//,1X,80A1)
      STOP 111
      END
```

Figure 60. Subroutine IDRO Program Listing (Concluded)

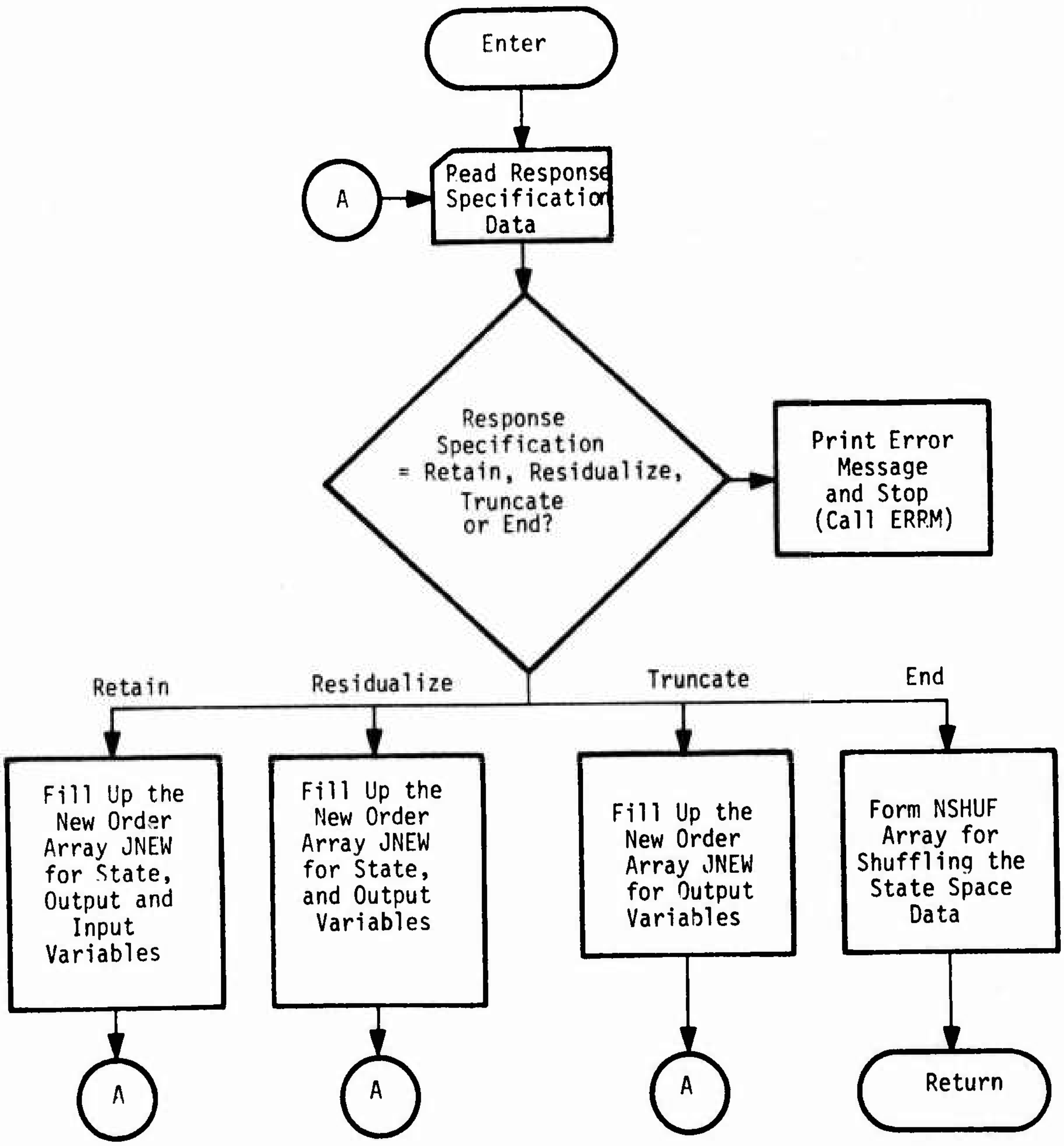

Figure 61. Subroutine RSDRD Flow Chart

```
      SUBROUTINE RSDRD(JNEWS,NSHUFS,JNEWO,NSHUFO,JNEWI,NSHUFI,
     1NX,NR,NU,NXRN,NXN,NXR,NRN,NRR,NRT,NUN,NXM,NRM,NUM,
     2IR,IW,IPRINT,IRS)
C
C     PURPOSE - TO READ REDUCTION AND SHUFFLING DATA
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        ERRM
C
C     ARGUMENTS LIST
C        JNEWS      OUTPUT     ARRAY FOR NEW ORDER OF STATES
C        JNEWO      OUTPUT     ARRAY FOR NEW ORDER OF OUTPUTS
C        JNEWI      OUTPUT     ARRAY FOR NEW ORDER OF INPUTS
C        NSHUFS     OUTPUT     SHUFFLING ARRAY FOR STATE
C        NSHUFO     OUTPUT     SHUFFLING ARRAY FOR OUTPUT
C        NSHUFI     OUTPUT     SHUFFLING ARRAY FOR INPUT
C        NX         INPUT      NUMBER OF STATES
C        NR         INPUT      NUMBER OF OUTPUTS
C        NU         INPUT      NUMBER OF INPUTS
C        NXRN       OUTPUT     NO OF STATES TO BE RETAINED AND RESIDUALI
C        NXN        OUTPUT     NUMBER OF REDUCED STATES
C        NRN        OUTPUT     NUMBER OF REDUCED OUTPUTS
C        NRR        OUTPUT     NO OF OUTPUTS TO BE RESIDUALIZED
C        NRT        OUTPUT     NO OF OUTPUTS TO BE TRUNCATED
C        NUN        OUTPUT     NUMBER OF REDUCED INPUTS
C        IR         INPUT      FILE NO FOR INPUT DATA BUFFER
C        IW         INPUT      FILE NO FOR LINE PRINTER
C        IPRINT     INPUT      PRINT CONTROL FLAG
C        IRS        OUTPUT     RESIDUALIZATION FLAG
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION JNEWS(NXM),NSHUFS(NXM)
      DIMENSION JNEWO(NRM),NSHUFO(NRM)
      DIMENSION JNEWI(NUM),NSHUFI(NUM)
      DIMENSION CARD(20)
      DATA HENDB,HRETA,HTATE,HXBBB/4HEND ,4HRETA,4HTATE,4HX   /
      DATA HNPUT,HUBBB,HRESI,HBSTA/4HNPUT,4HU   ,4HRESI,4H STA/
      DATA HRBBB,HTRUN,HUTPU,HSTAT/4HR   ,4HTRUN,4HUTPU,4HSTAT/
      IN=0
      IRS=0
      KS=0 $ KR=0 $ KU=0
      NXRN=0 $ NXN=0 $ NXR=0 $ NRN=0 $ NRR=0 $ NRT=0 $ NUN=0
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HRSDR,4HD   ,5,0,IW)
  120 CONTINUE
      READ(IR,140)CARD
  140 FORMAT(20A4)
      IF(CARD(1).EQ.HENDB)GO TO 420
      IRS=1
      IF(CARD(1).NE.HRETA)GO TO 240
      IF((CARD(3).NE.HTATE).AND.(CARD(3).NE.HNPUT).AND.
     1(CARD(3).NE.HUTPU))CALL ERRM(1,4HRSDR,4HD   ,5,0,IW)
  160 CONTINUE
C
C     READ SHUFFLE DATA FOR THE RETAINED SYSTEM VARIABLES
C
      READ(IR,180)HD,N
  180 FORMAT(A4,1X,I2)
      IF(HD.EQ.HENDB)GO TO 220
      IF(HD.NE.HXBBB)GO TO 200
C
C     FORM JNEWS ARRAY FOR STATES
C
      KS=KS+1
```

Figure 62. Subroutine RSDRD Program Listing

```
      JNEWS(KS)=N
      GO TO 160
  200 CONTINUE
      IF(HD.NE.HUBBB)GO TO 210
C
C     FORM JNEWI ARRAY FOR INPUTS
C
      KU=KU+1
      JNEWI(KU)=N
      GO TO 160
  210 CONTINUE
      IF(HD.NE.HRBBB)CALL ERRM(2,4HRSDR,4HD    ,5,0,IW)
C
C     FORM JNEWO ARRAY FOR OUTPUTS
C
      KR=KR+1
      JNEWO(KR)=N
      GO TO 160
  220 CONTINUE
      NXN=KS
      NUN=KU
      NRN=KR
      NXRN=NXN
      GO TO 120
  240 CONTINUE
      IF(CARD(1).NE.HRESI)GO TO 360
C
C     READ SHUFFLE DATA FOR THE RESIDUALIZED SYSTEM VARIABLES
C
      IF(CARD(4).NE.HSTAT)CALL ERRM(3,4HRSDR,4HD    ,5,0,IW)
  260 CONTINUE
      READ(IR,180)HD,N
      IF(HD.EQ.HENDB)GO TO 350
      IF(HD.NE.HXBBB)GO TO 280
C
C     FORM JNEWS ARRAY FOR STATES
C
      KS=KS+1
      JNEWS(KS)=N
      GO TO 260
  280 CONTINUE
      IF(HD.NE.HRBBB)CALL ERRM(4,4HRSDR,4HD    ,5,0,IW)
C
C     FORM JNEWO ARRAY FOR OUTPUTS
C
      KR=KR+1
      JNEWO(KR)=N
      GO TO 260
  350 CONTINUE
      NXRN=KS
      NXR=NXRN-NXN
      NRR=KR
      GO TO 120
  360 CONTINUE
      IF(CARD(1).NE.HTRUN)CALL ERRM(5,4HRSDR,4HD    ,5,0,IW)
      IF(CARD(3).NE.HBSTA)CALL ERRM(6,4HRSDR,4HD    ,5,0,IW)
C
C     READ SHUFFLE DATA FOR THE TRUNCATED SYSTEM VARIABLES
C
  400 CONTINUE
      READ(IR,180)HD,N
      IF(HD.EQ.HENDB)GO TO 410
      IF(HD.NE.HRBBB)CALL ERRM(7,4HRSDR,4HD    ,5,0,IW)
C
C     FORM JNEWO ARRAY FOR OUTPUTS
C
```

Figure 62. Subroutine RSDRD Program Listing (Continued)

```
      KR=KR+1
      JNEWO(KR)=N
      GO TO 400
  410 CONTINUE
      NRT=KR-NRR
      GO TO 120
  420 CONTINUE
      IF(IRS.EQ.0)RETURN
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HRSDR,4HD    ,5,0,IW)
      II=NXRN
C
C     FORM SHUFFLE ARRAY FOR STATES
C
      DO 470 I=1,NX
      DO 430 J=1,NXN
      JJ=J
      IF(I.EQ.JNEWS(JJ))GO TO 460
  430 CONTINUE
      IF(NXR.EQ.0)GO TO 450
      DO 440 J=1,NXR
      JJ=NXN+J
      IF(I.EQ.JNEWS(JJ))GO TO 460
  440 CONTINUE
  450 CONTINUE
      II=II+1
      NSHUFS(II)=I
      GO TO 470
  460 CONTINUE
      NSHUFS(JJ)=I
  470 CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HRSDR,4HD    ,5,0,IW)
C
C     FORM SHUFFLE ARRAY FOR OUTPUTS
C
      II=NRN
      IF(NRN.LE.0)II=NRT+NRR
      IF(II.EQ.0)IN=1
      DO 570 I=1,NR
      IF(NRN.LE.0)GO TO 520
      DO 510 J=1,NRN
      JJ=J
      IF(I.EQ.JNEWO(JJ))GO TO 560
  510 CONTINUE
      GO TO 550
  520 CONTINUE
      IF(NRR.EQ.0)GO TO 535
      DO 530 J=1,NRR
      JJ=J
      IF(I.EQ.JNEWO(JJ))GO TO 560
  530 CONTINUE
  535 CONTINUE
      IF(NRT.EQ.0)GO TO 550
      DO 540 J=1,NRT
      JJ=NRR+J
      IF(I.EQ.JNEWO(JJ))GO TO 560
  540 CONTINUE
  550 CONTINUE
      II=II+1
      NSHUFO(II)=I
      GO TO 570
  560 CONTINUE
      NSHUFO(JJ)=I
  570 CONTINUE
      IF(IN.EQ.1)NRN=NR
      IN=0
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HRSDR,4HD    ,5,0,IW)
```

Figure 62. Subroutine RSDRD Program Listing (Continued)

```
C
C     FORM SHUFFLE ARRAY FOR INPUTS
C
      II=NUN
      IF(II.EQ.0)IN=1
      DO 670  I=1,NU
      IF(NUN.LE.0)GO TO 640
      DO 630  J=1,NUN
      JJ=J
      IF(I.EQ.JNEWI(JJ))GO TO 660
  630 CONTINUE
  640 CONTINUE
      II=II+1
      NSHUFI(II)=I
      GO TO 670
  660 CONTINUE
      NSHUFI(JJ)=I
  670 CONTINUE
      IF(IN.EQ.1)NUN=NU
      IF(IPRINT.EQ.6)CALL DEBUG(5,4HRSDR,4HD    ,5,0,IW)
      IF(IPRINT.LT.6)GO TO 680
      WRITE(IW,675)NX,NXN,NR,NRN,NU,NUN
  675 FORMAT(1X,20(I2,1X))
      WRITE(IW,675)JNEWS,JNEWO,JNEWI
      WRITE(IW,675)NSHUFS,NSHUFO,NSHUFI
  680 CONTINUE
      RETURN
      END
```

Figure 62. Subroutine RSDRD Program Listing (Concluded)

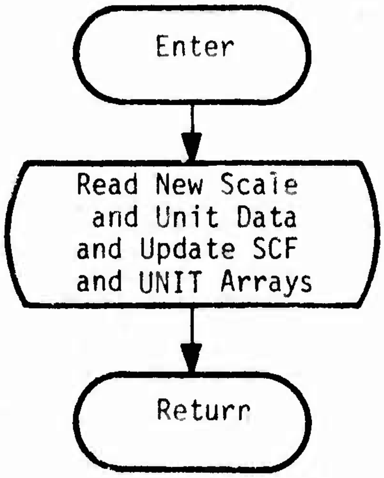

Figure 63. Subroutine SDRD Flow Chart

```
      SUBROUTINE SDRD(SCFS,UNITNS,UNITS,SCFO,UNITNO,UNITO,SCFI,UNITNI,
     1UNITI,NX,NR,NU,NXM,NRM,NUM,IR,IW,IPRINT)
C
C     PURPOSE - TO READ SCALE DATA
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C        ERRM
C
C     ARGUMENTS LIST
C        SCFS        OUTPUT     SCALING ARRAY FOR STATE
C        SCFO        OUTPUT     SCALING ARRAY FOR OUTPUT
C        SCFI        OUTPUT     SCALING ARRAY FOR INPUT
C        NX          INPUT      NUMBER OF STATES
C        NR          INPUT      NUMBER OF OUTPUTS
C        NU          INPUT      NUMBER OF INPUTS
C        IR          INPUT      FILE NO FOR INPUT DATA BUFFER
C        IW          INPUT      FILE NO FOR LINE PRINTER
C        IPRINT      INPUT      PRINT CONTROL FLAG
C     OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
      DIMENSION SCFS(NXM),UNITS(NXM,4),UNITNS(NXM,4)
      DIMENSION SCFO(NRM),UNITO(NRM,4),UNITNO(NRM,4)
      DIMENSION SCFI(NUM),UNITI(NUM,4),UNITNI(NUM,4)
      DIMENSION UN(4),UNN(4)
      DIMENSION CARD(20)
      DATA HENDB,HXBBB,HRBBB,HUBBB/4HEND ,4HX   ,4HR   ,4HU   /
      DATA HSCAL/4HSCAL/
      ISC=0
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSDRD,4H    ,5,0,IW)
C
C     INITIALIZE SCF ARRAY
C
      DO 140 I=1,NXM
  140 SCFS(I)=1.0
      DO 160 I=1,NRM
  160 SCFO(I)=1.0
      DO 180 I=1,NUM
  180 SCFI(I)=1.0
C
C     READ NEW SCALE AND UNIT DATA AND UPDATE SCF AND UNIT ARRAYS
C
  260 CONTINUE
      READ(IR,280)HD,N,SC,(UN(J),J=1,4),(UNN(J),J=1,4)
  280 FORMAT(A4,1X,I2,3X,E14.6,6X,4A4,4X,4A4)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HSDRD,4H    ,5,0,IW)
      IF(HD.EQ.HENDB)RETURN
      ISC=1
      IF(HD.NE.HXBBB)GO TO 320
C
C     FOR STATES
C
      SCFS(N)=SC
      DO 300 J=1,4
      UNITNS(N,J)=UNN(J)
  300 UNITS(N,J)=UNN(J)
      GO TO 260
  320 CONTINUE
      IF(HD.NE.HRBBB)GO TO 360
C
C     FOR OUTPUTS
C
      SCFO(N)=SC
      DO 340 J=1,4
```

Figure 64. Subroutine SDRD Program Listing

```
      UNITNO(N,J)=UNN(J)
  340 UNITO(N,J)=UNN(J)
      GO TO 260
  360 CONTINUE
      IF(HD.NE.HUBBB)CALL ERRM(2,4HSDRD,4H    ,6,0,IW)
C
C     FOR INPUTS
C
      SCFI(N)=SC
      DO 380 J=1,4
      UNITNI(N,J)=UNN(J)
  380 UNITI(N,J)=UNN(J)
      GO TO 260
      END
```

Figure 64. Subroutine SDRD Program Listing (Concluded)

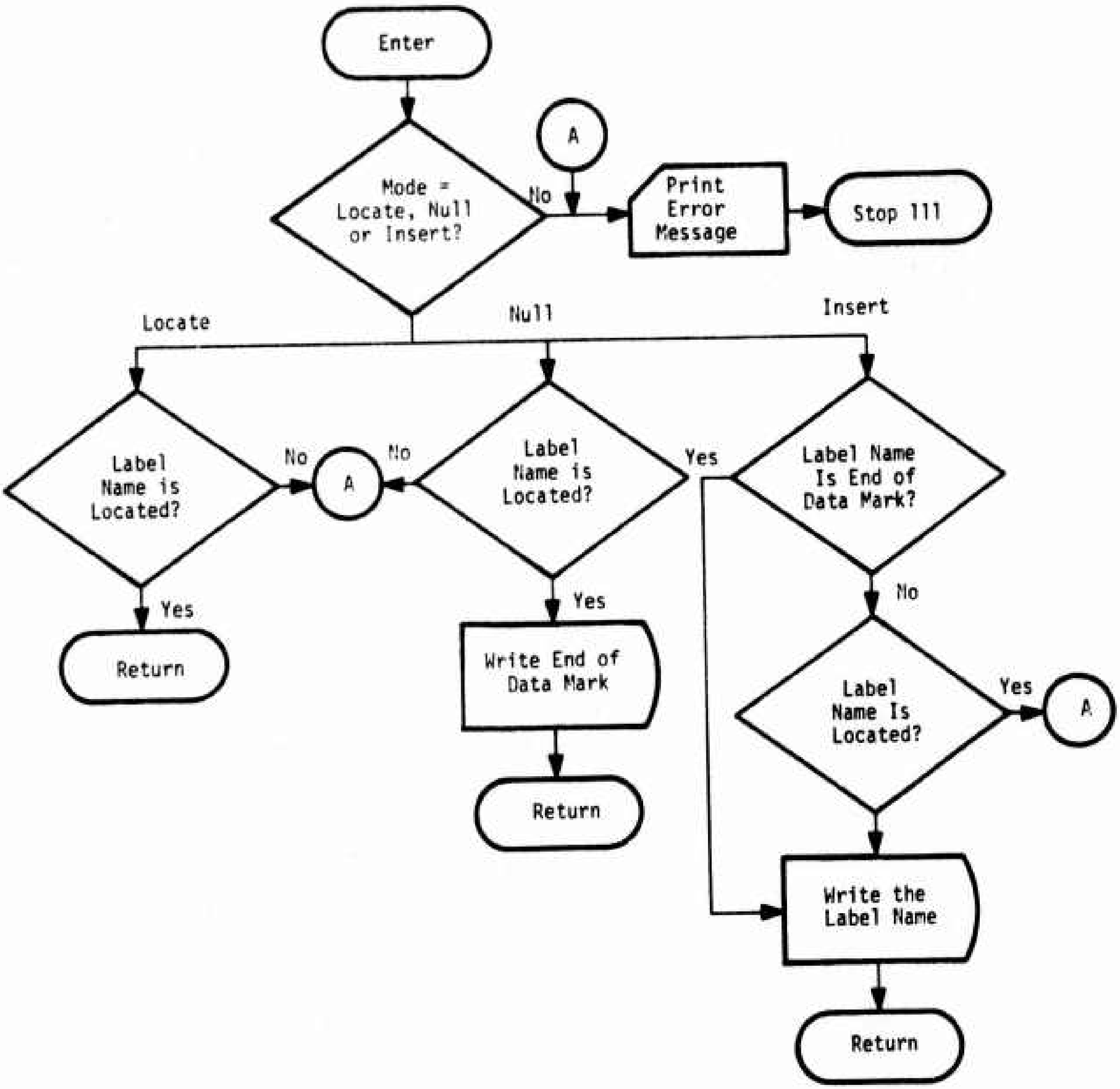

Figure 65. Subroutine FILE Flow Chart

```
      SUBROUTINE FILE(NFILE,MODE,NAME)
C
C     PURPOSE - TO POSITION THE DATA FILE
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENT LIST
C        NFILE                 FILE NUMBER OF THE DISK FILE
C        MODE                  MODE PARAMETER (LOCATE,INSERT,NULL)
C        NAME                  LABEL NAME
C
      DIMENSION NAME(20),LABEL(20),LAST(20)
      INTEGER HOOLR
      DATA LOCATE,INSERT,NULL/4HLOCA,4HINSE,4HNULL/
      DATA HOOLR/4H$$$$/
      IW=9
      DO 100 I=1,20
  100 LAST(I)=HOOLR
      IF(MODE.EQ.LOCATE)GO TO 140
      IF(MODE.EQ.INSERT)GO TO 120
      IF(MODE.EQ.NULL)GO TO 140
C
C     PRINT ERROR MESSAGE
C
      WRITE(IW,110)
  110 FORMAT(1H1,//,1X,45HMODE OF OPERATION FOR DATA FILE NOT SPECIFIED)
      STOP 111
C
C     CHECK IF END OF DATA MARK IS BEING INSERTED
C
  120 CONTINUE
      DO 130 I=1,20
      IF(NAME(I).NE.LAST(I))GO TO 140
  130 CONTINUE
C
C     WRITE END OF DATA MARK AND ALSO AN END OF FILE
C
      WRITE(NFILE)(NAME(I),I=1,20)
      ENDFILE NFILE
      REWIND NFILE
      RETURN
C
C     CHECK IF LABEL ON FILE MATCHES WITH NAME
C
  140 CONTINUE
      REWIND NFILE
  150 CONTINUE
      READ(NFILE)(LABEL(I),I=1,20)
      DO 160 I=1,20
      IF(LABEL(I).NE.NAME(I))GO TO 170
  160 CONTINUE
      GO TO 220
C
C     CHECK IF LABEL IS THE END OF DATA MARK
C
  170 CONTINUE
      DO 180 I=1,20
      IF(LABEL(I).NE.LAST(I))GO TO 200
  180 CONTINUE
      IF(MODE.EQ.INSERT)GO TO 210
C
C     PRINT ERROR MESSAGE
C
```

Figure 66. Subroutine FILE Program Listing

```
      WRITE(IW,190)NAME,NFILE
  190 FORMAT(1H1,//,1X,20A4,//,1X,29HCANNOT BE FOUND ON DATA FILE ,I2)
      STOP 111
C
C     POSITION THE FILE TO THE BEGINNING OF NEXT RECORD
C
  200 CONTINUE
      READ(NFILE)
      GO TO 150
C
C     WRITE NAME ON THE FILE
C
  210 BACKSPACE NFILE
      WRITE(NFILE)(NAME(I),I=1,20)
      RETURN
  220 CONTINUE
      IF(MODE.EQ.INSERT)GO TO 230
      IF(MODE.EQ.LOCATE)RETURN
C
C     WRITE END OF DATA MARK
C
      BACKSPACE NFILE
      WRITE(NFILE)(LAST(I),I=1,20)
      RETURN
C
C     PRINT ERROR MESSAGE
C
  230 CONTINUE
      WRITE(IW,240)NAME,NFILE
  240 FORMAT(1H1,//,1X,20A4,//,1X,21HALREADY ON DATA FILE ,I2)
      STOP 111
      END
```

Figure 66. Subroutine FILE Program Listing (Concluded)

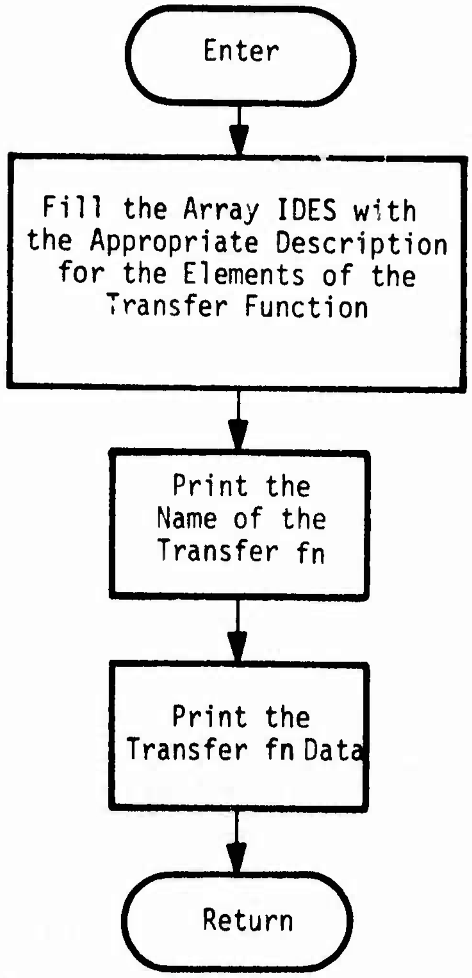

Figure 67. Subroutine TPR Flow Chart

```
      SUBROUTINE TPR(H,NE,NEM,NAME,T,IW)
C
C     PURPOSE - TO PRINT TRANSFER FUNCTION DATA
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C          H          INPUT     TRANSFER FUNCTION
C          NE         INPUT     NO OF ELEMENTS OF THE TRANSFER FN
C          NEM        INPUT     MAXIMUM NO OF ELEMENTS OF THE TRANSFER FN
C          NAME       INPUT     NAME OF THE TRANSFER FN
C          T          INPUT     SAMPLE TIME
C          IW         INPUT     FILE NO FOR LINE PRINTER
C
C     LABELLED COMMON LIST
C          IDES       LOCAL     ARRAY FOR DESCRIPTION OF THE TRANSFER FN
C          M1         LOCAL     CONSTANT
C          I          LOCAL     INDEX
C          K          LOCAL     INDEX
C
      DIMENSION H(2,NEM),NAME(3)
      COMMON /SC1/ IDES(6,3),M1,I,K
C
C     CHECK FOR DIMENSION ERROR
C
      IF((NEM.NE.6).OR.(NE.GT.NEM))GO TO 260
C
C     FILL THE ARRAY IDES WITH THE APPROPRIATE DESCRIPTION FOR THE
C     ELEMENTS OF THE TRANSFER FUNCTION
C
      IF(T.NE.0.0)GO TO 120
      IDES(1,1)=4HS**5 $ IDES(1,2)=4H TER $ IDES(1,3)=4HM
      IDES(2,1)=4HS**4 $ IDES(2,2)=4H TER $ IDES(2,3)=4HM
      IDES(3,1)=4HS**3 $ IDES(3,2)=4H TER $ IDES(3,3)=4HM
      IDES(4,1)=4HS**2 $ IDES(4,2)=4H TER $ IDES(4,3)=4HM
      IDES(5,1)=4HS**1 $ IDES(5,2)=4H TER $ IDES(5,3)=4HM
      IDES(6,1)=4HS**0 $ IDES(6,2)=4H TER $ IDES(6,3)=4HM
      GO TO 140
  120 CONTINUE
      IDES(1,1)=4HZ**5 $ IDES(1,2)=4H TER $ IDES(1,3)=4HM
      IDES(2,1)=4HZ**4 $ IDES(2,2)=4H TER $ IDES(2,3)=4HM
      IDES(3,1)=4HZ**3 $ IDES(3,2)=4H TER $ IDES(3,3)=4HM
      IDES(4,1)=4HZ**2 $ IDES(4,2)=4H TER $ IDES(4,3)=4HM
      IDES(5,1)=4HZ**1 $ IDES(5,2)=4H TER $ IDES(5,3)=4HM
      IDES(6,1)=4HZ**0 $ IDES(6,2)=4H TER $ IDES(6,3)=4HM
  140 CONTINUE
C
C     PRINT THE NAME OF THE TRANSFER FN
C
      IF(T.EQ.0.0)WRITE(IW,160)NAME
  160 FORMAT(//,1X,3A4)
      IF(T.NE.0.0)WRITE(IW,180)NAME,T
  180 FORMAT(//,1X,3A4,3H(T=,G14.6,1H))
C
C     PRINT THE TRANSFER FN
C
      M1=6-NE+1
      WRITE(IW,200)((IDES(I,K),K=1,3),I=M1,6)
  200 FORMAT(//,18X,5(3A4,2X))
      WRITE(IW,220)(H(1,I),I=1,NE)
  220 FORMAT(/,1X,9HNUMERATOR,6X,5G14.6)
      WRITE(IW,240)(H(2,I),I=1,NE)
  240 FORMAT(/,1X,11HDENOMINATOR,4X,5G14.6)
```

Figure 68. Subroutine TPR Program Listing

```
      RETURN
C
C     PRINT ERROR MESSEGE
C
  260 CONTINUE
      WRITE(IW,280)
  280 FORMAT(1H1,//,1X,42HDIMENSION ERROR DETECTED BY SUBROUTINE TPR)
      STOP 111
      END
```

Figure 68. Subroutine TPR Program Listing (Concluded)

```
      SUBROUTINE HPR(CARD,IW)
C
C     PURPOSE - TO PRINT HEADING FOR SYSTEM LABEL NAMES
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C          CARD       INPUT      SYSTEM LABEL NAME
C          IW         INPUT      FILE NO FOR LINE PRINTER
C
      DIMENSION CARD(20)
      INTEGER CARD
      WRITE(IW,120)
  120 FORMAT(1H1,//,20X,88(1H*))
      WRITE(IW,140)
  140 FORMAT(20X,1H*,86X,1H*)
      WRITE(IW,160)CARD
  160 FORMAT(20X,1H*,2X,20A4,4X,1H*)
      WRITE(IW,140)
      WRITE(IW,180)
  180 FORMAT(20X,88(1H*))
      RETURN
```

Figure 69. Subroutine HPR Program Listing

```
      SUBROUTINE IDPR(IR,IW)
C
C     PURPOSE - TO PRINT INPUT DATA
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C         IR          INPUT      FILE NO FOR INPUT DATA BUFFER
C         IW          INPUT      FILE NO FOR LINE PRINTER
C
      DIMENSION CARD(20)
      REWIND IR
  120 CONTINUE
      READ(IR,140)CARD
  140 FORMAT(20A4)
      IF(EOF(IR))200,160
  160 CONTINUE
      WRITE(IW,180)CARD
  180 FORMAT(1X,20A4)
      GO TO 120
  200 CONTINUE
      REWIND IR
      RETURN
      END
```

Figure 70. Subroutine IDPR Program Listing

```
      SUBROUTINE MPRS(A,MXR,MXC,NR,NC,T,ITITLE)
C
C     PURPOSE - TO PRINT MATRIX DATA
C     ANALYSTS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        A           INPUT      MATRIX DATA
C        MXR         INPUT      MAXIMUM NO OF ROWS
C        MXC         INPUT      MAXIMUM NO OF COLUMNS
C        NR          INPUT      NUMBER OF ROWS
C        NC          INPUT      NUMBER OF COLUMNS
C        T           INPUT      SAMPLE TIME
C        ITITLE      INPUT      TITLE OR NAME OF THE MATRIX
C
      DIMENSION A(MXR,MXC)
C
C     BEGINNING OF COLUMN SIZE LOOP
C
      JC =
  100 CONTINUE
      IC = JC + 1
      JC = JC + 1
      IF (JC .GT. NC)  JC = NC
C
C     BEGINNING OF ROW SIZE LOOP
C
      JR=0
  150 CONTINUE
C
C     PRINT MATRIX NAME AND SIZE
C
      IF(T.EQ.0.0) WRITE(9,80) ITITLE,NR,NC
   80 FORMAT (//8H MATRIX ,A4,16X,7HSIZE = ,I2,3H X ,I2)
      IF(T.NE.0.0) WRITE(9,90) ITITLE,T,NR,NC
   90 FORMAT (//8H MATRIX ,A4,3H(T=,E10.4,1H),2X,7HSIZE = ,I2,3H X ,I2)
C
C     PRINT COLUMN INDEX
C
      WRITE (9, 160) (K, K = IC, JC)
  160 FORMAT (//8X,10(2X,I3,7H-COLUMN))
      WRITE(9,170)
  170 FORMAT(/)
      IR=JR+1
      JR=JR+100
      IF(JR.GT.NR) JR=NR
      DO 180  I=IR,JR
C
C     PRINT ROW INDEX AND MATRIX DATA
C
      WRITE (9, 190) I,(A(I,J), J = IC,JC)
  190 FORMAT (1X,I3,4H-ROW,1X,10(F12.4))
  180 CONTINUE
C
C     END OF ROW SIZE LOOP
C
      IF(JR.LT.NR)GO TO 150
C
C     END OF COLUMN SIZE LOOP
C
      IF (JC .LT. NC)  GO TO 100
C
C     RETURN TO CALLING PROGRAM
```

Figure 71. Subroutine MPRS Program Listing

```
C                                                                       MPRS  65
      RETURN                                                            MPRS  66
      END                                                               MPRS  67
```

Figure 71. Subroutine MPRS Program Listing (Concluded)

```
      SUBROUTINE MPRS1(A,NRM,NCM,NR,NC,MHEAD)
C
C     PURPOSE - TO PRINT LSA MATRIX DATA
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        A          INPUTT    MATRIX DATA
C        NRM        INPUT     MAXIMUM NUMBER OF ROWS
C        NCM        INPUT     MAXIMUM NUMBER OF COLUMNS
C        NR         INPUT     NUMBER OF ROWS
C        NC         INPUT     NUMBER OF COLUMNS
C        MHEAD      INPUT     MATRIX TITLE OR NAME
C
      DIMENSION A(NRM,NCM)
      COMMON /INOUT/ IR,IW,IPRINT,JN,JQ,INSERT,LOCATE,NULL,MARK(20)
      IF((IPRINT.NE.3).AND.(IPRINT.LT.5))RETURN
C
C     WRITE NAME AND SIZE OF THE MATRIX
C
      WRITE(IW,80)MHEAD,NR,NC
   80 FORMAT(//,10X,A10,16HMATRIX , SIZE = ,I2,3H X ,I2)
      JC=0
  100 IC=JC+1
      JC=JC+7
      IF(JC.GT.NC)JC=NC
      KC=JC-IC+1
C
C     WRITE COLUMN HEADINGS
C
  150 WRITE(IW,160)(K,K=IC,JC)
  160 FORMAT(//,8X,7(2X,I3,7H-COLUMN,3X))
      WRITE(IW,170)
  170 FORMAT(/)
      DO 18  I=1,NR
C
C     WRITE ROW HEADINGS
C
  180 WRITE(IW,190)I,(A(I,J),J=IC,JC)
  190 FORMAT(1X,I3,4H-ROW,1X,7(E15.7))
  320 IF(JC.LT.NC)GO TO 100
      WRITE(IW,170)
      RETURN
      END
```

Figure 72. Subroutine MPRS1 Program Listing

```
      SUBROUTINE ZERO(A,NRM,NCM)
C
C     PURPOSE - TO ZERO THE ELEMENTS OF A MATRIX
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C         A          OUTPUT     MATRIX DATA
C         NRM        INPUT      MAXIMUM NUMBER OF ROWS
C         NCM        INPUT      MAXIMUM NUMBER OF COLUMNS
C
      DIMENSION A(NRM,NCM)
      DO 120 I=1,NRM
      DO 120 J=1,NCM
  120 A(I,J)=0.0
      RETURN
      END
```

Figure 73. Subroutine ZERO Program Listing

```
      SUBROUTINE INPT(A,II,JJ)
C
C     PURPOSE - TO READ NON ZERO ELEMENTS OF A MATRIX
C
C     ARGUMENTS LIST
C        A          OUTPUT    MATRIX DATA
C        II         INPUT     MAXIMUM NO OF ROWS
C        JJ         INPUT     MAXIMUM NO OF COLUMNS
C
      DIMENSION A(II,JJ),ID(5),JD(5),YD(5)
    1 READ(5,2)(ID(I),JD(I),YD(I),I=1,5)
    2 FORMAT(5(2I2,E12.5))
      IF(ID(1))10,10,3
    3 DO 6 L=1,5
      IF(ID(L))4,1,4
    4 CONTINUE
      I=ID(L)
      J=JD(L)
      A(I,J)=YD(L)
    6 CONTINUE
      GO TO 1
   10 CONTINUE
      RETURN
      END
```

Figure 74. Subroutine INPT Program Listing

```
      SUBROUTINE INPT1(A,NRM,NCM,NR,NC,IR)
C
C     PURPOSE - TO READ ISA MATRIX DATA
C     ANALISTS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        A          OUTPUT     MATRIX DATA
C        NRM        INPUT      MAXIMUM NUMBER OF ROWS
C        NCM        INPUT      MAXIMUM NUMBER OF COLUMNS
C        NR         INPUT      NUMBER OF ROWS
C        NC         INPUT      NUMBER OF COLUMNS
C        IR         INPUT      FILE NO FOR INPUT DATA BUFFER
C
      DIMENSION A(NRM,NCM)
      READ(IR,120)((A(I,J),J=1,NC),I=1,NR)
  120 FORMAT(6G10.3)
      RETURN
      END
```

Figure 75. Subroutine INPT1 Program Listing

```
      SUBROUTINE DEBUG(N,A1,A2,N1,N2,IW)
C
C     PURPOSE - TO PRINT DEBUGGING MESSAGE
C     ANALISTS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        N          INPUT      POSITION OF EXECUTION
C        A1         INPUT      NAME OF THE SUBROUTINE
C        A2         INPUT      NAME OF THE SUBROUTINE (CONTINUED)
C        N1         INPUT      PRIMARY OVERLAY NO
C        N2         INPUT      SECONDARY OVERLAY NO
C        IW         INPUT      FILE NO FOR LINE PRINTER
C
      WRITE(IW,120)N,A1,A2,N1,N2
  120 FORMAT(//,1X,27HEXECUTION ENTERED POSITION ,I2,1X,
     111HSUBROUTINE ,2A4,1X,12HIN OVERLAY (,I1,1H,,I1,1H))
      RETURN
      END
```

Figure 76. Subroutine DEBUG Program Listing

```
      SUBROUTINE ERRM(N,A1,A2,N1,N2,IW)
C
C     PURPOSE - TO PRINT ERROR MESSEGE
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        N          INPUT       POSITION OF EXECUTION
C        A1         INPUT       NAME OF THE SUBROUTINE
C        A2         INPUT       NAME OF THE SUBROUTINE (CONTINUED)
C        N1         INPUT       PRIMARY OVERLAY NO
C        N2         INPUT       SECONDARY OVERLAY NO
C        IW         INPUT       FILE NO FOR LINE PRINTER
C
      WRITE(IW,120)N,A1,A2,N1,N2
  120 FORMAT(1H1,//,1X,27HERROR DETECTED AT POSITION ,I2,1X,
     111HSUBROUTINE ,2A4,1X,12HIN OVERLAY (,I1,1H,,I1,1H))
      STOP 111
      END
```

Figure 77. Subroutine ERRM Program Listing

```
      SUBROUTINE DERRM(M1,M2,M3,M4,MS1,MS2,MS3,MS4,N1,N2,A1,A2,IW)
C
C     PURPOSE - TO PRINT ERROR MESSEGE WHEN DIMENSIONS FOR
C     SCRATCH ARRAYS IS NOT SUFFICIENT
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        M1          INPUT       ACTUAL DIMENSION FOR SCRATCH ARRAY S1
C        M2          INPUT       ACTUAL DIMENSION FOR SCRATCH ARRAY S2
C        M3          INPUT       ACTUAL DIMENSION FOR SCRATCH ARRAY S3
C        M4          INPUT       ACTUAL DIMENSION FOR SCRATCH ARRAY S4
C        MS1         INPUT       MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C        MS2         INPUT       MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C        MS3         INPUT       MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C        MS4         INPUT       MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C        N1          INPUT       PRIMARY OVERLAY NO
C        N2          INPUT       SECONDARY OVERLAY NO
C        A1          INPUT       NAME OF THE SUBROUTINE
C        A2          INPUT       NAME OF THE SUBROUTINE (CONTINUED)
C        IW          INPUT       FILE NO FOR LINE PRINTER
C
      DIMENSION M(4),MS(4)
      M(1)=M1 $ M(2)=M2 $ M(3)=M3 $ M(4)=M4
      MS(1)=MS1 $ MS(2)=MS2 $ MS(3)=MS3 $ MS(4)=MS4
      WRITE(IW,240)N1,N2,A1,A2
  240 FORMAT(1H1,//,1X,28HDIMENSION ERROR IN OVERLAY (,I1,1H,,I1,1H),
     113HIN SUBROUTINE,2X,2A4)
      DO 260 I=1,4
      WRITE(IW,250)I,MS(I),M(I)
  250 FORMAT(//,1X,15HDIMENSION FOR S,I1,2X,7HACTUAL=,I5,2X,
     19HREQUIRED=,I5)
  260 CONTINUE
      STOP 111
      END
```

Figure 78. Subroutine DERRM Program Listing

```
      SUBROUTINE DERRMS(M1,M2,M3,M4,MS1,MS2,MS3,MS4,N1,N2,A1,A2,IW)
C
C     PURPOSE - TO PRINT ERROR MESSEGE WHEN SYSTEM DIMENSION
C     ARE NOT SUFFICIENT
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        M1          INPUT      ACTUAL DIMENSION
C        M2          INPUT      ACTUAL DIMENSION
C        M3          INPUT      ACTUAL DIMENSION
C        M4          INPUT      ACTUAL DIMENSION
C        MS1         INPUT      MAXIMUM DIMENSION
C        MS2         INPUT      MAXIMUM DIMENSION
C        MS3         INPUT      MAXIMUM DIMENSION
C        MS4         INPUT      MAXIMUM DIMENSION
C        N1          INPUT      PRIMARY OVERLAY NO
C        N2          INPUT      SECONDARY OVERLAY NO
C        A1          INPUT      NAME OF THE SUBROUTINE
C        A2          INPUT      NAME OF THE SUBROUTINE (CONTINUED)
C        IW          INPUT      FILE NO FOR LINE PRINTER
C
      DIMENSION M(4),MS(4),A(4)
      DATA A/4HNXM ,4HNRM ,4HNUM ,4HNYM /
      M(1)=M1 $ M(2)=M2 $ M(3)=M3 $ M(4)=M4
      MS(1)=MS1 $ MS(2)=MS2 $ MS(3)=MS3 $ MS(4)=MS4
      WRITE(IW,240)N1,N2,A1,A2
  240 FORMAT(1H1,//,1X,28HDIMENSION ERROR IN OVERLAY (,I1,1H,,I1,1H),
     113HIN SUBROUTINE,2X,2A4)
      DO 260  I=1,4
      WRITE(IW,250)A(I),MS(I),M(I)
  250 FORMAT(//,1X,10HDIMENSION ,A4,2X,7HACTUAL=,I5,2X,
     19HREQUIRED=,I5)
  260 CONTINUE
      STOP 111
      END
```

Figure 79. Subroutine DERRMS Program Listing

```
      SUBROUTINE SHIFT(NN,VN,DES,UNIT,NNN,VNN,DESN,UNITN,N,NM,IW,IPRINT)
C
C     PURPOSE - TO SHIFT CONTENTS OF OLD ARRAYS NN,VN,DES,UNIT
C     INTO NEW ARRAYS NNN,VNN,DESN,UNITN
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DEBUG
C
C     ARGUMENTS LIST
C        NN          INPUT       OLD NUMBER ARRAY
C        VN          INPUT       OLD VARIABLE NAME ARRAY
C        DES         INPUT       OLD DESCRIPTION ARRAY
C        UNIT        INPUT       OLD UNIT ARRAY
C        NNN         OUTPUT      NEW NUMBER ARRAY
C        VNN         OUTPUT      NEW VARIABLE NAME ARRAY
C        DESN        OUTPUT      NEW DESCRIPTION ARRAY
C        UNITN       OUTPUT      NEW UNIT ARRAY
C        N           INPUT       NUMBER OF SYSTEM VARIABLES
C        NM          INPUT       MAX NO OF SYSTEM VARIABLES
C        IW          INPUT       FILE NO FOR LINE PRINTER
C        IPRINT      INPUT       PRINT CONTROL FLAG
C
      DIMENSION NN(NM),VN(NM,2),DES(NM,10),UNIT(NM,4)
      DIMENSION NNN(NM),VNN(NM,2),DESN(NM,10),UNITN(NM,4)
      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSHIF,4HT    ,5,0,IW)
      DO 140 I=1,N
      NNN(I)=NN(I)
      DO 120 J=1,2
  120 VNN(I,J)=VN(I,J)
      DO 130 J=1,10
  130 DESN(I,J)=DES(I,J)
      DO 140 J=1,4
  140 UNITN(I,J)=UNIT(I,J)
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HSHIF,4HT    ,5,0,IW)
      RETURN
      END
```

Figure 80. Subroutine SHIFT Program Listing

```
      SUBROUTINE TDINVR(ISOL,IDSOL,NR,NC,A,MRA,KWA,DET)
C
C     PURPOSE - TO INVERT A NONSINGULAR MATRIX OR
C               TO SOLVE LINEAR EQUATIONS
C
C     ARGUMENTS LIST
C        ISOL                   TDINVR SOLUTION INDICATOR
C                               ISOL=1 IF INVERSE FOUND OR
C                               EQUATIONS SOLVED
C        IDSOL                  DETERMINANT OVERFLOW INDICATOR
C                               IDSOL=1 IF DETERMINANT CALCULATION
C                               DID NOT OVERFLOW
C        NR                     NUMBER OF ROWS
C        NC                     NUMBER OF COLUMNS
C        A                      MATRIX TO BE INVERTED OR CONTAINING
C                               THE LINEAR EQUATION COEFFICIENTS
C        MRA                    MAXIMUM NUMBER OF ROWS OF A
C        KWA                    SCRATCH ARRAY WHEN INVERTING
C        DET                    VALUE OF THE DETERMINANT
C
      DIMENSION A(1),KWA(1)
      IR=NR
      ISOL =1
      IDSOL=1
      IF(NR) 61,61,11
   11 IF(IR-MRA)12,12,61
   12 IC=IABS(NC)
      IF(IC-IR) 13,14,14
   13 IC=IR
   14 IBMP=1
      JBMP=MRA
      KBMP=JBMP+IBMP
      NES=IR*JBMP
      NET=IC*JBMP
      IF(NC)  15,61,16
   15 MDIV= JBMP+1
      IRIC=IR-IC
      GO TO 17
   16 MDIV=1
   17 MAD =MDIV
      MSER=1
      KSER=IR
      MZ  =1
      DET=1.0
   18 PIV=0.0
      I=MSER
   19 IF(I-KSER)          20,20,23
   20 IF(ABS(A(I))-PIV)22,22,21
   21 PIV=ABS(A(I))
      IP=I
   22 I=I+IBMP
      GO TO 19
   23 IF(PIV)   24,62,24
   24 IF(NC)   26,25,25
   25 I=IP-((IP-1)/JBMP)*JBMP
      J=MSER-((MSER-1)/JBMP)*JBMP
      JJ=MSER/KBMP+1
      II=JJ+(IP-MSER)
      KWA(JJ)=II
      GO TO 27
   26 I=IP
      J=MSER
   27 IF(IP-MSER) 61,31,28
```

Figure 81. Subroutine TDINVR Program Listing

```
 28 IF(J-NET)  29,29,30                                          TDINVR65
 29 PSTO=A(I)                                                    TDINVR66
    A(I)=A(J)                                                    TDINVR67
    A(J)=PSTO                                                    TDINVR68
    I=I+JBMP                                                     TDINVR69
    J=J+JBMP                                                     TDINVR70
    GO TO 28                                                     TDINVR71
 30 DET=-DET                                                     TDINVR72
 31 PSTO=A(MSER)                                                 TDINVR73
    DET=DET*PSTO                                                 TDINVR74
 35 PSTO=1.0/PSTO                                                TDINVR75
    A(MSER)=1.0                                                  TDINVR76
    I=MDIV                                                       TDINVR77
 36 IF(I-NET)    37,37,39                                        TDINVR78
 37 A(I)=A(I)*PSTO                                               TDINVR79
    I=I+JBMP                                                     TDINVR80
    GO TO 36                                                     TDINVR81
 39 IF(MZ-KSER) 40,40,145                                        TDINVR82
 40 IF(MZ-MSER) 41,44,41                                         TDINVR83
 41 I=MAD                                                        TDINVR84
    J=MDIV                                                       TDINVR85
    PSTO=A(MZ)                                                   TDINVR86
    IF(PSTO) 142,44,142                                          TDINVR87
142 A(MZ)=0.0                                                    TDINVR88
 42 IF(J-NET)    43,43,44                                        TDINVR89
 43 A(I)=A(I)-A(J)*PSTO                                          TDINVR90
    J=J+JBMP                                                     TDINVR91
    I=I+JBMP                                                     TDINVR92
    GO TO 42                                                     TDINVR93
 44 MAD=MAD+IBMP                                                 TDINVR94
    MZ=MZ+IBMP                                                   TDINVR95
    GO TO 39                                                     TDINVR96
145 KSER=KSER+JBMP                                               TDINVR97
    IF(KSER-NES) 46,46,53                                        TDINVR98
 46 MSER=MSER+KBMP                                               TDINVR99
    IF(NC) 48,47,47                                              TDINV100
 47 MDIV=MDIV+IBMP                                               TDINV101
    MZ=((MSER-1)/JBMP)*JBMP+1                                    TDINV102
    MAD=1                                                        TDINV103
    GO TO 52                                                     TDINV104
 48 MDIV=MDIV+KBMP                                               TDINV105
    IF(IRIC) 50,49,50                                            TDINV106
 49 MZ=MSER+IBMP                                                 TDINV107
    GO TO 51                                                     TDINV108
 50 MZ=((MSER-1)/JBMP)*JBMP+1                                    TDINV109
 51 MAD=MZ+JBMP                                                  TDINV110
 52 GO TO 18                                                     TDINV111
 53 IF(NC) 65,54,54                                              TDINV112
 54 JR=IR                                                        TDINV113
 55 IF(JR) 61,65,56                                              TDINV114
 56 IF(KWA(JR)-JR) 61,60,57                                      TDINV115
 57 K=(JR-1)*JBMP                                                TDINV116
    J=K+IR                                                       TDINV117
    L=(KWA(JR)-1)*JBMP+IR                                        TDINV118
 58 IF(J-K)    61,60,59                                          TDINV119
 59 PSTO=A(L)                                                    TDINV120
    A(L)=A(J)                                                    TDINV121
    A(J)=PSTO                                                    TDINV122
    J=J-IBMP                                                     TDINV123
    L=L-IBMP                                                     TDINV124
    GO TO 58                                                     TDINV125
 60 JR=JR-1                                                      TDINV126
    GO TO 55                                                     TDINV127
 61 ISOL=3                                                       TDINV128
    GO TO 65                                                     TDINV129
 62 DET=0.0                                                      TDINV130
```

Figure 81. Subroutine TDINVR Program Listing (Continued)

```
      ISOL=?                                                            TDINV131
      IDSOL=1                                                           TDINV132
      GO TO 65                                                          TDINV133
   63 ISOL = 2                                                          TDINV134
      IDSOL = 2                                                         TDINV135
   65 RETURN                                                            TDINV136
      END                                                               TDINV137
```

Figure 81. Subroutine TDINVR Program Listing (Concluded)

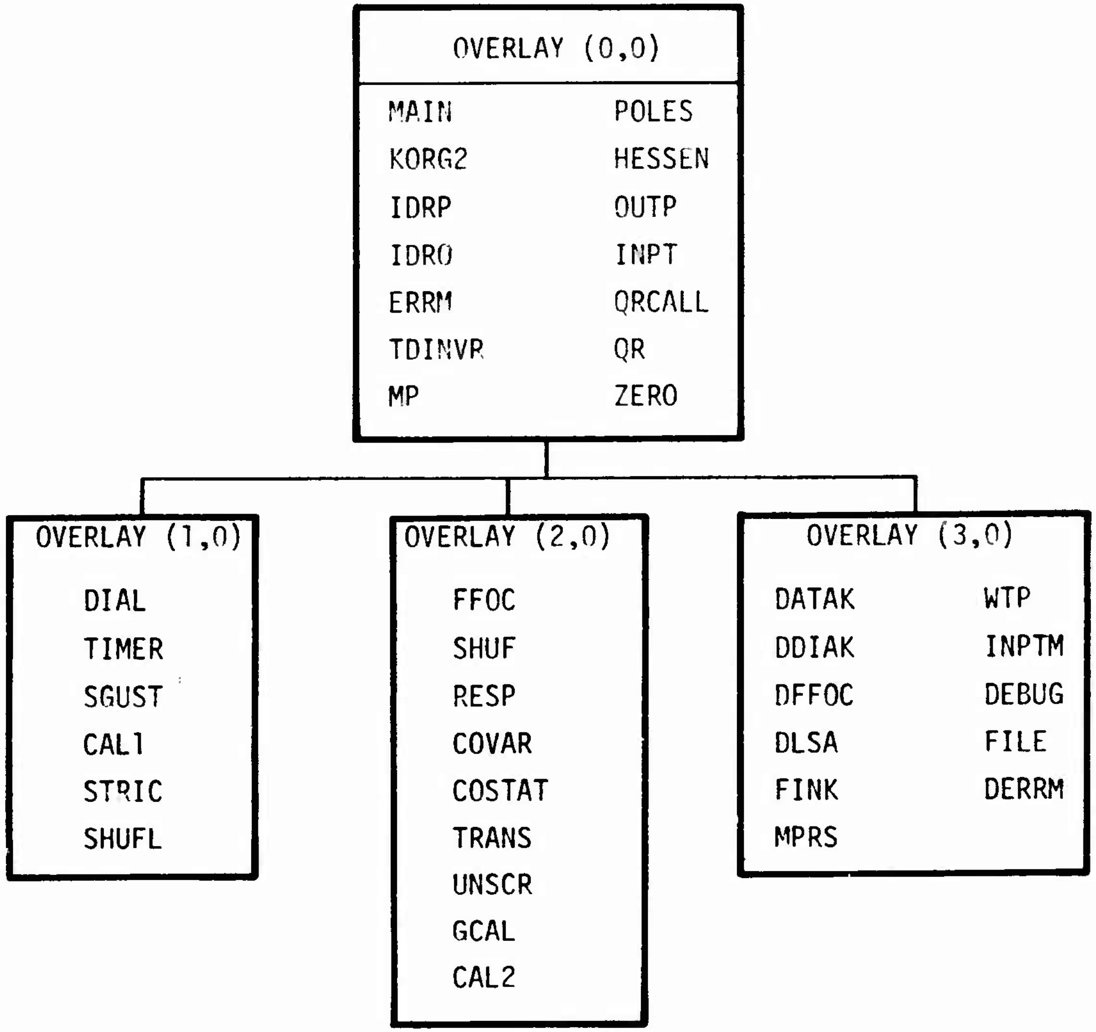

Figure 82. Overlay Structure and Subroutines in KONPACT-2

```
      OVERLAY(KON2,0,0)
      PROGRAM MAIN(MINPUT,INPUT,TAPE7=MINPUT,TAPE4=INPUT,
     1ODATA,OUTPUT,TAPE8=ODATA,TAPE9=OUTPUT,SCRTCH,TAPE5=SCRTCH,
     2FDATA,DDATA,TAPE1=FDATA,TAPE6=DDATA,SDSTP,TAPE2=SDSTP)
C
C     PURPOSE - TO SET UP MAXIMUM DIMENSIONS
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        KORG2
C
C     LABELLED COMMON LIST
C        NXM                 MAXIMUM NUMBER OF STATES
C        NRM                 MAXIMUM NUMBER OF OUTPUTS
C        NUM                 MAXIMUM NUMBER OF INPUTS
C        CODE                PROGRAM CODE WORD (DIAK,FFOC,LSA)
C        MS1                 MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C        MS2                 MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C        MS3                 MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C        MS4                 MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C
      COMMON /INF/ NXM,NRM,NUM,CODE,MS1,MS2,MS3,MS4
C
C     MAXIMUM SYSTEM DIMENSIONS
C
      NXM=50 $ NRM=70 $ NUM=20
C
C     MAXIMUM SCRATCH ARRAY DIMENSIONS
C
      MS1=08500 $ MS2=17000 $ MS3=00001 $ MS4=00001
C
C     *** NOTE *** SCRATCH ARRAY DIMENSIONS IN PROGRAM DATAK
C     SHOULD BE CHANGED
C
C     CALL KONPACT ORGANIZING SUBROUTINE
C
      CALL KORG2
      STOP
      END
```

Figure 83. Program MAIN Program Listing

```
      OVERLAY(KON2,1,0)
      PROGRAM DIAK
C DOUBLY-ITERATIVE ALGORITHM FOR SOLVING ALGEBRAIC RICCATI EQUATION
C   THIS PROGRAM COMPUTES QUADRATIC CONTROLLERS AND/OR COMPUTES COVARIAN
C        TIME RESPONSES FOR SYSTEMS MODELED AS
C
C                  XDOT = F*X + G1*U + G2*ETA
C        AND
C                  R = H*X + D*U
C        WITH
C                  J = E(R#*Q*R)
C
      DIMENSION F(40,40),G1(40,6),G2(40,2),A(40,40),AN(40,40),E(40,40)
      DIMENSION Q(40,40), WR(40,40),EP(40,40),P(40,40),H(40,40),D(40,6)
      DIMENSION AK(6,40),PI(40,40),DOD(6,6),KWA(40),W(6,40),W1(6,40)
      DIMENSION QQ(40,40),RR(90),AM(40,40),BK(6,40),X(40),DX(40),DX1(40)
      DIMENSION XI(40,2),XLDXL(40,2),GN(40,2),GS(40,2),R(8000),IPLR(80)
      DIMENSION ITITL(80),YMAX(80),YMIN(80),CL(2,1),SCAL(80),NEWY(80)
      DIMENSION NORD(40),QR(40,40),IUNIT(80)
      COMMON A,E,Q,AN,WR,QR,EP,P,PI
      EQUIVALENCE (F(1),P(1)),(H(1),PI(1)),(AM(1),EP(1))
      EQUIVALENCE (R(1),E(1))
C   DIMENSIONS OF THE ABOVE ARRAYS ARE DEFINED BELOW, CHANGE BOTH SIMULT
C        SEE DOCUMENTATION FOR DEFINITIONS OF ARRAY DIMENSIONS
C       MX>NX
C       MR>NR
C       MU>NU
C       MN>NN
C       MXR>NOP
C       MPOIN>(NOP + 1)*(T/ST)
      MR=40
      MXR=80
      MX=40
      MU=6
      MN=2
      MPOIN=8000
C   CONVERGENCE TEST FACTOR
      EE=0.001
C   ITERATION COUNTER
C   RUN COUNTER
      IRUN=1
C   READ AND PRINT ID
      READ(5,1274) IDATE,NAME1,NAME2
 1274 FORMAT(3A10)
      WRITE(9,1275) IDATE,NAME1,NAME2
1275  FORMAT(1H1,7X,13HTODAY#S DATE ,A10,5X,16HIDENTIFICATION  ,2A10//)
C   READ NUMBER OF VARIABLES BEING PLOTTED
      READ(5,28) NOP
   28 FORMAT(40I2)
C        IF NOP = 0, SKIP TO STATEMENT 70
      IF(NOP.EQ.0) GO TO 70
C   READ PLOTTING PARAMETERS - THEY ARE FIXED FOR ALL RUNS
C        IPLR = ARRAY OF PLOTTING VARIABLE NOS. - READ IN ORDER
C        ITITL = CORRESPONDING ARRAY OF LABELS
C        IUNIT = CORRESPONDING ARRAY OF UNIT LABELS
C        YMAX,YMIN = CORRESPONDING ARRAYS OF DESIGNATED MAX AND MIN VALUES
C        SCAL = CORRESPONDING ARRAY OF SCALE FACTORS
      READ(5,1272) ((IPLR(I),ITITL(I),IUNIT(I),YMIN(I),YMAX(I),SCAL (I))
     1,I=1,NOP)
1272  FORMAT(I2,2X,A10,2X,A10,2X,G11.3,2X,G11.3,2X,G11.3)
C   DEFINE PLOTTING SCALES - FIXED FOR ALL RUNS
C        IF YMIN AND YMAX ARE 0, USE COMPUTED MAX AND MIN (NEWY=1)
C        IF SCAL = 0, USE SCALE FACTOR OF 1
```

Figure 84. Program DIAK Program Listing

```
      DO 1 I=1,NOP
      NEWY(I)=0
      IF(YMIN(I).EQ.0..AND.YMAX(I).EQ.0.) NEWY(I)=1
      IF(SCAL (I).EQ.0.) SCAL (I)=1.
    1 CONTINUE
C   READ AND PRINT PLOTTING TIME PARAMETERS - FIXED FOR ALL RUNS
C      T = TOTAL PLOTTING TIME
C      DT = SAMPLING INTERVAL
C      ST = PLOTTING SAMPLING INTERVAL
C      T1 = FIRST DELAY IN GUST PROFILE
C      T2 = SECOND DELAY IN GUST PROFILE
      READ(5,1278) T,DT,ST,T1,T2
 1278 FORMAT(5G12.4)
C   PRINT PLOTTING PARAMETERS
      WRITE(9,1279) T,DT,ST,T1,T2
 1279 FORMAT(1H0/7X,31H TIME RESPONSES PLOTTING TIME =,G12.4/22X,18H SAM
     1PLE INTERVAL =,G12.4/22X,27H PLOTTING SAMPLE INTERVAL =,G12.4/
     222X,19H FIRST DELAY TIME =,G12.4/22X,20H SECOND DELAY TIME =,G12.4
     3//)
      WRITE(9,1285) ((IPLR(I),ITITL(I),IUNIT(I),YMIN(I),YMAX(I),SCAL (I)
     1),I=1,NOP)
1285  FORMAT(7X,18HPLOTTING VARIABLES//2X,86HRESPONSE NUMBER  RESPONSE V
     1ARIABLE  RESPONSE UNITS  MIN SCALE  MAX SCALE  SCALE FACTOR//(2X,
     2I10,9X,A10,9X,A10,3X,G11.3,2X,G11.3,3X,G11.3))
C   READ AND PRINT MAX NO. OF INNER AND OUTER LOOP ITERATIONS
   70 READ(5,28) IMAX,ITER,ITERO
      WRITE(9,4002)IMAX,ITER,ITERO
 4002 FORMAT(////7X,37H MAX NUMBER OF INNER-LOOP ITERATIONS I3,37H MAX N
     1UMBER OF OUTER-LOOP ITERATIONS I3/7X,67H MAX NUMBER OF ITERATIONS
     2ON ELIMINATING CONTROL SURFACE FEEDBACKS I3//)
C   DEFINITION OF PROGRAM OPTIONS
C  INPD=1  COMPLETELY NEW DATA
C  INPD=2  CHANGE SELECTED QUADRATIC WEIGHTS ONLY - USE SOME GAINS IN  S
C  INPD=3  CHANGE SELECTED QUADRATIC WEIGHTS ONLY WITH OPTION FOR NEW GA
C  INPD=4  CHANGE SELECTED DATA
C  INPD=5  CHANGE SELECTED DATA IN MEASUREMENT MATRIX, QUADRATIC WEIGHTS
C          OPTION FOR NEW GAINS
C  INPK=1  NEW INPUT GAINS
C  INPK=2  NEW STARTING ROUTINE GAINS
C  INPK=3  USE GAINS IN STORAGE
C  INPK=4  USE INPUT GAINS IN STORAGE
C  NCONT=0  DON*T COMPUTE OPTIMAL GAINS - USE INPUT GAINS AND DATA IN CO
C           AND TIME RESPONSE ANALYSIS ONLY
C  NCONT=1  COMPUTE OPTIMAL GAINS
C  NCONT=2  DO AUTOMATIC SELECTION OF Q ON CONTROL RATES
C  SEE SUBROUTINE TIMER FOR PLOTTING OPTIONS USING NPLOT, NPRIN, NSTEP,
C  NOCOV=1  NO COVARIANCE ANALYSIS
C  NOCOV=2  COVARIANCE ANALYSIS
C  NOCOV=3  SKIP CORRELATION ANALYSIS
C   READ AND PRINT PROGRAM OPTIONS
      READ(5,28) NOCOV,NSTEP,NRAND,NPRIN ,NPLOT
      READ(5,28) INPK
      INPD=1
      READ(5,28) NCONT
      WRITE(9,37) INPD,INPK,NCONT,NOCOV,NSTEP,NRAND,NPRIN ,NPLOT
   37 FORMAT(1H1/7X,23HNEW PROBLEM WITH INPD =,I3,2X,6HINPK =,I3,2X,
     17HNCONT =,I3/7X,7HNOCOV =,I3/7X,7HNSTEP =,I3,2X,7HNRAND =,I3/7X,
     27HNPRIN =,I3,2X,7HNPLOT =,I3//)
 1210 CONTINUE
C   READ FLIGHT CONDITION ID
      READ(5,1270)IFLT
1270  FORMAT(A10)
C   PRINT FLIGHT CONDITION ID AND RUN NO.
      WRITE(9,1271) IFLT, IRUN
1271  FORMAT(1H1/7X,18H FLIGHT CONDITION A10,5X,3HRUN,I3)
C   READ AND PRINT SYSTEM PARAMETERS
```

Figure 84. Program DIAK Program Listing (Continued)

```
C      NX = NO. OF STATES
C      NR = NO. OF RESPONSES
C      NU = NO. OF CONTROLS
C      NN = NO. OF DISTURBANCE INPUTS
C   NSCRR = RESPONSE STARTING CONTROL RATE RESPONSES
C    PARAMETERS FOR PLOTTING
C      NF = NO. OF FEEDBACK STATES = NX - NO. OF DISTURBANCE STATES (NOT
C       NG = NO. OF GUST INPUTS
C       NCS = NO. OF COMMAND INPUTS = NO. OF COMMAND STATES
C       NGLG = NO. OF GUST LIFT GROWTH STATES
      READ(5,28) NX,NR,NU,NN,NF,NG,NCS,NGLG,NSCRR
      WRITE(9,4003) NX,NR,NU,NN,NF,NG,NCS,NGLG,NSCRR
 4003 FORMAT(//7X,18H ORDER OF SYSTEM =I3/7X,22H NUMBER OF RESPONSES =I3
     1/7X,21H NUMBER OF CONTROLS =I3/7X,31H NUMBER OF DISTURBANCE INPUTS
     2 =,I3/8X,27HNUMBER OF FEEDBACK STATES =,I3/7X,24H NUMBER OF GUST I
     3NPUTS =,I3/7X,27H NUMBER OF COMMAND STATES =,I3/7X,36H NUMBER OF G
     4UST LIFT GROWTH STATES =,I3/7X,43H CONTROL RATE RESPONSES START WI
     5TH RESPONSEI3//)
C   NC IS THE NUMBER OF UPPER TRIANGULAR ELEMENTS IN P
      NC=(NX*(NX+1))/2
C
C ZERO ARRAYS
C
C      RIGHT HAND PARAMETERS DEFINED BELOW
      DO 8020 I=1,MX
      DO 8013 J=1,MX
      F  (I,J)=0.
      A  (I,J)=0.
      AN (I,J)=0.
      E  (I,J)=0.
      EP (I,J)=0.
      PI (I,J)=0.
      AM(I,J)=0.
 8013 CONTINUE
      DO 8014 J=1,NU
 8014 G1(I,J)=0.
      DO 8015 J=1,NN
      XI(I,J)=0.
      CL(J,1)=0.
 8015 G2(I,J)=0.
      DO 8016 J=1,2
8016  XLDXL(I,J)=0.
 8020 CONTINUE
      DO 3060 I=1,NR
      DO 3061 J=1,NR
 3061 QQ(I,J)=0.
      DO 3062 J=1,MX
 3062 H(I,J)=0.
      DO 3060 J=1,NU
 3060 D(I,J)=0.
C    READ DATA FOR THIS RUN
1240   IF(INPD.GT.1) GO TO 53
C       IF INPD = 1  (NEW DATA), READ ORDERING OF STATES
C       NORD = ARRAY OF THE ORDER OF STATES
      READ(5,28) (NORD(I),I=1,NX)
      WRITE(9,67) (NORD(I),I=1,NX)
67    FORMAT(//7X,22H STATES ARE ORDERED AS//(7X,20I4)//)
53    CONTINUE
C    READ CHANGES IN F,G1,G2 (ROW, COLUMN, ELEMENT VALUE)
C       F = STABILITY MATRIX (OPEN LOOP)
C       G1 = CONTROL INPUT MATRIX
C       G2 = DISTURBANCE INPUT MATRIX
C       WHERE
C               XDOT = F*X + G1*U + G2*ETA
C       IF INPD>1 (CHANGES TO EXISTING DATA), ROW AND COLUMNS CORRESPOND
C          RE-ORDERED STATES
```

Figure 84. Program DIAK Program Listing (Continued)

```
      CALL INPT (F,MX,MX)
      CALL INPT(G1,MX,MU)
      CALL INPT (G2,MX,MN)
      IF(INPD.GT.1) GO TO 54
C   IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
      CALL SHUFL(F,MX,MX,NX,NX,1,1,NORD,Q,MX)
      CALL SHUFL(G1,MX,MU,NX,NU,1,0,NORD,Q,MX)
      CALL SHUFL(G2,MX,MN,NX,NN,1,0,NORD,Q,MX)
54    CONTINUE
C   PRINT F,G1,G2
      WRITE(9,20)
      CALL MP(MX,MX,NX,NX,F)
      WRITE(9,21)
      CALL MP(MX,MU,NX,NU,G1)
      WRITE(9,22)
   22 FORMAT(1H1/7X,10H G2 MATRIX//)
      CALL MP(MX,MN,NX,NN,G2)
   20 FORMAT(1H1/7X,10H F  MATRIX//)
   21 FORMAT(1H1/7X,10H G1 MATRIX//)
C   READ CHANGES IN XI AND XLDXL
C      XI = INITIAL STATE VALUES IN SIMULATION
C      XLDXL = STATE AND STATE RATE LIMITS
      CALL INPT(XI,MX,MN)
      CALL INPT(XLDXL,MX,2)
      IF(INPD.GT.1) GO TO 55
C   IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
      CALL SHUFL(XI,MX,MN,NX,NN,1,0,NORD,Q,MX)
      CALL SHUFL(XLDXL,MX,2,NX,2,1,0,NORD,Q,MX)
   55 CONTINUE
C   READ CL = STEP GUST AND COMMAND INPUT LEVELS
      CALL INPT(CL,MN,1)
C   PRINT XI, XLDXL, CL
      WRITE(9,1276)
 1276 FORMAT(1H1/7X,24HINITIAL CONDITION MATRIX//)
      CALL MP(MX,MN,NX,NN,XI)
      WRITE(9,1277)
 1277 FORMAT(1H1/7X,31HSTATE LIMIT - RATE LIMIT MATRIX//)
      CALL MP(MX,2,NX,2,XLDXL)
      WRITE(9,1273)
 1273 FORMAT(1H1/7X,20HCOMMAND LEVEL MATRIX//)
      CALL MP(MN,1,NN,1,CL)
C   READ IN CHANGES IN H AND D
C      H = STATE-RESPONSE OUTPUT MATRIX
C      D = CONTROL-RESPONSE OUTPUT MATRIX
C
C              R = H*X + D*U
C      WHERE
C
   73 CALL INPT (H,MR,MX)
      CALL INPT(D,MR,MU)
      IF(INPD.GT.1) GO TO 1250
C   IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)
      CALL SHUFL(H,MR,MX,NR,NX,0,1,NORD,Q,MX)
C   READ CHANGES IN M (AM)
3      AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY
C      WHERE
C              Y = M*X
C
 1250 CALL INPT (AM,MX,MX)
      IF(INPD.GT.1) GO TO 56
C   IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)
      CALL SHUFL(AM,MX,MX,NX,NX,0,1,NORD,Q,MX)
56    CONTINUE
C   PRINT H, D, AM
      WRITE(9,23)
      CALL MP(MR,MX,NR,NX,H)
```

Figure 84. Program DIAK Program Listing (Continued)

```
      WRITE(9,24)
      CALL MP(MR,MU,NR,NU,D)
      WRITE(9,29)
29    FORMAT(1H1/7X,10H M  MATRIX//)
      CALL MP(MX,MX,NX,NX,AM)
   23 FORMAT(1H1/7X,10H H  MATRIX//)
   24 FORMAT(1H1/7X,10H D  MATRIX//)
C   CHECK GAINS INPUT OPTION
 1230 GO TO (3001,3002,3010,3011),INPK
C   NEW INPUT GAINS
C      BK = INPUT GAINS MATRIX
C      WHERE
C             U = BK*X      (WHEN COMPUTING OPTIMAL GAINS)
C      OR
C             U = BK*Y = BK*AM*X    (WHEN COMPUTING RESPONSES ONLY)
C
C      ZERO AND READ BK
 3001 DO 3003 I=1,NU
      DO 3003 J=1,MX
 3003 BK(I,J)=0.
      CALL INPT (BK,MU,MX)
      IF(NCONT.EQ.0) GO TO 57
C   IF NCONT>0, RE-ORDER STATES (BECAUSE U = BK*X)
      CALL SHUFL(BK,MU,MX,NU,NX,0,1,NORD,Q,MX)
57    CONTINUE
C   PRINT BK
      WRITE(9,30)
   30 FORMAT(1H1/7X,19H INPUT GAINS MATRIX//)
      CALL MP(MU,MX,NU,NX,BK)
C   SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
      GO TO 1220
C   USE STARTING ROUTINE (STRIC) TO COMPUTE STARTING GAINS - AS A LAST R
C      BK = -G1*(W(T))!    (A! MEANS INVERSE OF MATRIX A)
C      WHERE
C            W(T) = INTEGRAL(0,BT)OF(EXP(F*T)*G1*G1#*EXP(F#*T))DT
C      FOR AN ARBITRARY TIME BT
 3002 CALL STRIC(F,G1,A,AN,E,Q,MF,NU,MX,MU)
      CALL TDINVR(ISOL,IDSOL,NF,NF,AN,MX,KWA,DET)
      IF((ISOL+IDSOL)-2) 3004,3004,3005
C   (W(T))! IS NO GOOD - GO TO NEXT RUN - BUT FIRST, READ REMAINING DATA
C      THIS RUN AND CHECK TO SEE IF THE NEXT RUN IS SOLVABLE - THE START
C      GAINS MAY NOT BE GOOD - IF SO, STOP
 3005 WRITE(9,3006)
 3006 FORMAT(1H1/7X,31H INVERSE OF W(T) DOES NOT EXIST/7X,18H CHECK NEXT
     1PROBLEM//)
      CALL INPT (QQ,MR,MR)
      READ(5,1215) IDUM
      IF(IDUM.GT.0) STOP 77
      READ(5,28) INPD,INPK
      IF(INPK.EQ.1) GO TO 1216
      IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4))GO TO 1216
      WRITE(9,3008)
 3008 FORMAT(/7X,51HNEW PROBLEM NOT SOLVABLE WITHOUT NEW STARTING GAINS/
     1/)
      STOP 11
C   DEFINE BK
C      AN = (WCT))!
C
 3004 DO 3009 I=1,NU
      DO 3009 J=1,NF
      BK(I,J)=0.
      DO 3009 K=1,NF
 3009 BK(I,J)=BK(I,J)-G1(K,I)*AN(K,J)
C   PRINT BK
      WRITE(9,31)
   31 FORMAT(1H1/7X,22H STARTING GAINS MATRIX//)
```

Figure 84. Program DIAK Program Listing (Continued)

```
      CALL MP(MU,MX,NU,NF,BK)
C   SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
      GO TO 1220
C   USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS
C       DEFINE BK = AK
 3010 WRITE(9,33)
   33 FORMAT(1H1/7X,28H USE GAINS MATRIX IN STORAGE//)
C   SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
      GO TO 1220
C   USE INPUT GAINS IN STORAGE - BK = BK
 3011 WRITE(9,34)
   34 FORMAT(1H1/7X,34H USE INPUT GAINS MATRIX IN STORAGE//)
C   READ CHANGES IN QUADRATIC WEIGHTS FOR PERFORMANCE INDEX
C
C          J = E(R#*Q*R)
C       WHERE Q IS THE MATRIX OF QUADRATIC WEIGHTS
C       QQ = Q
 1220 CONTINUE
      CALL INPT(QQ,MR,MR)
      NQ=1
   81 CONTINUE
C   PRINT QQ
      WRITE(9,36)
   36 FORMAT(1H1/7X,27H QUADRATIC WEIGHTING MATRIX//)
      CALL MP(MR,MR,NR,NR,QQ)
C   IF NCONT = 0 (NO OPTIMAL CONTROL COMPUTATIONS), SKIP TO STATEMENT 89
C       RESPONSE COMPUTATIONS
      IF(NCONT.EQ.0) GO TO 893
C   CALCULATE A,E,Q FOR RICCATI EQUATION 0 = PA + A#P + Q - PEP
C   W = D#*Q
      DO 4 I=1,NU
      DO 4 J=1,NR
      W(I,J)=0.
      DO 4 K=1,NR
    4 W(I,J)=W(I,J)+D(K,I)*QQ(K,J)
C   DQD = D#*Q*D
      DO 5 I=1,NU
      DO 5 J=1,NU
      DQD(I,J)=0.
      DO 5 K=1,NR
    5 DQD(I,J)=DQD(I,J)+W(I,K)*D(K,J)
C   INVERT DQD - DQD = (D#*Q*D)I
      IF(NU-1)302,302,301
  302 DQD(1,1)=1./DQD(1,1)
      GOTO 303
  301 CONTINUE
      CALL TDINVR(ISOL,IDSOL,NU,NU,DQD,MU,K,WA,DET)
      IF((ISOL+IDSOL)-2)6,6,7
C   (D#*Q*D)I DOES NOT EXIST - GO TO NEXT RUN
    7 WRITE(9,35)
   35 FORMAT(1H1/7X,30H INVERSE OF DQD DOES NOT EXIST//7X,19H CHECK NEXT
     1 PROBLEM//)
      GO TO 1200
    6 CONTINUE
  303 CONTINUE
C   W1 = D#*Q*H
      DO 8 I=1,NU
      DO 8 J=1,NX
      W1(I,J)=0.
      DO 8 K=1,NR
    8 W1(I,J)=W1(I,J)+W(I,K)*H(K,J)
C   W = (D#*Q*D)I*D#*Q*H
C       STORE W FOR OPTIMAL CONTROL COMPUTATION
      DO 9 I=1,NU
      DO 9 J=1,NX
      W(I,J)=0.
```

Figure 84. Program DIAK Program Listing (Continued)

```
      DO 9 K=1,NU                                                       DIAK 395
    9 W(I,J)=W(I,J)+DQD(I,K)*W1(K,J)                                    DIAK 396
C     AN = F - G1*(D#*Q*D)I*D#*Q*H                                      DIAK 397
C        AN = A OF EQUATION 0 = A#P + PA + Q - PEP                      DIAK 398
      DO 10 I=1,NX                                                      DIAK 399
      DO 10 J=1,NX                                                      DIAK 400
      AN(I,J)=F(I,J)                                                    DIAK 401
      DO 10 K=1,NU                                                      DIAK 402
   10 AN(I,J)=AN(I,J)-G1(I,K)*W(K,J)                                    DIAK 403
C     Q = -H#*Q*D*(D#*Q*D)I*D#*Q*H                                      DIAK 404
      DO 12 I=1,NX                                                      DIAK 405
      DO 12 J=1,NX                                                      DIAK 406
      Q(I,J)=0.                                                         DIAK 407
      DO 12 K=1,NU                                                      DIAK 408
   12 Q(I,J)=Q(I,J)-W1(K,I)*W(K,J)                                      DIAK 409
C     E = Q*H                                                           DIAK 410
      DO 13 I=1,NR                                                      DIAK 411
      DO 13 J=1,NX                                                      DIAK 412
      E(I,J)=0.                                                         DIAK 413
      DO 13 K=1,NR                                                      DIAK 414
   13 E(I,J)=E(I,J)+QQ(I,K)*H(K,J)                                      DIAK 415
C     Q = H#*Q*H - H#*Q*D*(D#*Q*D)I*D#*Q*H                              DIAK 416
C        Q = Q OF EQUATION 0 = A#P + PA + Q - PEP                       DIAK 417
      DO 14 I=1,NX                                                      DIAK 418
      DO 14 J=I,NX                                                      DIAK 419
      DO 15 K=1,NR                                                      DIAK 420
   15 Q(I,J)=Q(I,J)+H(K,I)*E(K,J)                                       DIAK 421
   14 Q(J,I)=Q(I,J)                                                     DIAK 422
C     W1 = (D#*Q*D)I*G1#                                                DIAK 423
      DO 16 I=1,NU                                                      DIAK 424
      DO 16 J=1,NX                                                      DIAK 425
      W1(I,J)=0                                                         DIAK 426
      DO 16 K=1,NU                                                      DIAK 427
   16 W1(I,J)=W1(I,J)+DQD(I,K)*G1(J,K)                                  DIAK 428
C     E = G1*(D#*Q*D)I*G1#                                              DIAK 429
C        E = E OF EQUATION 0 = A#P + PA + Q - PEP                       DIAK 430
      DO 17 I=1,NX                                                      DIAK 431
      DO 17 J=I,NX                                                      DIAK 432
      E(I,J)=0.                                                         DIAK 433
      DO 18 K=1,NU                                                      DIAK 434
   18 E(I,J)=E(I,J)+G1(I,K)*W1(K,J)                                     DIAK 435
   17 E(J,I)=E(I,J)                                                     DIAK 436
C     PRINT AN,E,Q                                                      DIAK 437
      WRITE(9,32)                                                       DIAK 438
   32 FORMAT(1H1/7X,36HSTARTING MATRICES FOR PA+A*P+Q-PEP=0//)          DIAK 439
      WRITE(9,25)                                                       DIAK 440
      CALL MP(MX,MX,NX,NX,AN)                                           DIAK 441
      WRITE(9,26)                                                       DIAK 442
      CALL MP(MX,MX,NX,NX,E)                                            DIAK 443
      WRITE(9,27)                                                       DIAK 444
      CALL MP(MX,MX,NX,NX,Q)                                            DIAK 445
   25 FORMAT(//7X,10H A   MATRIX//)                                     DIAK 446
   26 FORMAT(1H1/7X,10H E   MATRIX//)                                   DIAK 447
   27 FORMAT(1H1/7X,10H Q   MATRIX//)                                   DIAK 448
C     DUMP F,H, AND AM ON DISC TO CONSERVE STORAGE                      DIAK 449
C        P, PI, AND EP USE STORAGE EQUIVALENT TO THESE MATRICES         DIAK 450
      REWIND 2                                                          DIAK 451
      WRITE(2) F                                                        DIAK 452
      WRITE(2) H,AM                                                     DIAK 453
      ITERC=0                                                           DIAK 454
C     CHECK GAINS INPUT OPTION                                          DIAK 455
      GO TO (3000,3000,2050,3000),INPK                                  DIAK 456
C     FOR ALL OPTIONS EXCEPT INPK = 3 (USE AK IN STORAGE), AK = BK      DIAK 457
 3000 DO 7010 I=1,NU                                                    DIAK 458
      DO 7010 J=1,NX                                                    DIAK 459
 7010 AK(I,J)=BK(I,J)                                                   DIAK 460
```

Figure 84. Program DIAK Program Listing (Continued)

```
 2050 CONTINUE
C    A = F+ G1*K
C        A = CLOSED LOOP STABILITY MATRIX
      DO 7011 I=1,NX
      DO 7011 J=1,NX
      A(I,J)=F(I,J)
      DO 7011 K=1,NU
 7011 A(I,J)=A(I,J)+G1(I,K)*AK(K,J)
C    H= H+ D*K
C        H IS NOW CLOSED LOOP STATE-RESPONSE OUTPUT MATRIX
      DO 7012 I=1,NR
      DO 7012 J=1,NX
      DO 7012 K=1,NU
 7012 H(I,J)=H(I,J)+D(I,K)*AK(K,J)
C    COMPUTE (H+D*K)#*Q*(H+D*K)
C        P = Q*(H+D*K)
      DO 7013 I=1,NR
      DO 7013 J=1,NX
      P(I,J)=0.
      DO 7013 K=1,NR
 7013 P(I,J)=P(I,J)+QQ(I,K)*H(K,J)
C        EP = (H+D*K)#*P = (H+D*K)#*Q*(H+D*K)
      DO 7014 I=1,NX
      DO 7014 J=1,NX
      EP(I,J)=0.
      DO 7014 K=1,NR
 7014 EP(I,J)=EP(I,J)+H(K,I)*P(K,J)
C    SOLVE FOR INITIAL RICCATI MATRIX P FROM
C        0 = A#*P+P*A+ (H+D*K)#*Q*(H+D*K)
C
C        VIA SUBROUTINE CAL
3        P IS WORKING MATRIX HERE - RICCATI MATRIX RETURNS IN EP
      CALL CAL1(A,EP,P,KWA,NX,MX,IMAX,1,IERR,EE)
      IF(IERR.EQ.0) GO TO 875
C    ERROR ENCOUNTERED IN CAL - GO TO NEXT RUN
      WRITE(9,38)
   38 FORMAT(1H1/7X,27H INITIAL GAINS ARE UNSTABLE//7X,19H CHECK NEXT PRO
     1OBLEM//)
      READ(5,1215) IDUM
      IF(IDUM.GT.0) STOP 77
      READ(5,28) INPD,INPK
      IF(INPK.EQ.1) GO TO 1216
      IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216
C    NEXT RUN NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP
      WRITE(9,3008)
      STOP 11
C    SET P = EP, INITIALIZE PI = 0
  875 DO 876 I=1,NX
      DO 876 J=1,NX
      PI(I,J)=0.
  876 P(I,J)=EP(I,J)
C    UPDATE A AND Q MATRICES FOR NEXT ITERATION
C        A = AN - E*P
C        Q = Q+ P*E*P
C        TO SOLVE FOR P FROM
C
C            0 = A#*P + P*A + Q
C
C        VIA SUBROUTINE CAL
C    AFTER SOLVING FOR SECOND P, SOLVE FOR DIFFERENCES IN P BETWEEN ITERA
C        THUS INITIALIZE DIFFERENCES AND CONVERGENCE CRITERIA
C        P IS DIFFERENCE AND PI IS THE TOTAL RICCATI MATRIX
C        INITIALLY PI IS ZERO
      DO 100 I=1,NX
      DO 100 J=1,NX
      EP(I,J)=0.
```

Figure 84. Program DIAK Program Listing (Continued)

```
      DO 101 K=1,NX
  101 EP(I,J)=EP(I,J)+E(I,K)*P(K,J)
  100 A(I,J)=AN(I,J)-EP(I,J)
      DO 102 I=1,NX
      DO 102 J=I,NX
      DO 103 K=1,NX
  103 Q(I,J)=Q(I,J)+P(I,K)*EP(K,J)
  102 Q(J,I)=Q(I,J)
      FEE=EF
 1000 CONTINUE
      DO 2010 I=1,NX
      DO 2010 J=1,NX
 2010 PI(I,J)=P(I,J)+PI(I,J)
      CALL SECOND(TT)
      WRITE(9,3055) TT
 3055 FORMAT(//7X,6HTIME =,F10.5//)
C     CALL CAL -P IS AGAIN WORKING MATRIX - RICCATI MATRIX RETURNS IN Q
      CALL CAL1(A,Q,P,KWA,NX,MX,IMAX,1,IERR,FEE)
      FEE=EF*10.
      CALL SECOND(TT)
      WRITE(9,3055) TT
      IF(IERR.EQ.0) GO TO 874
C     ERROR ENCOUNTERED IN CAL - GO TO NEXT RUN
      WRITE(9,39)
   39 FORMAT(1H1/7X,30H RICCATI SOLUTION IS DIVERGING//7X,19H CHECK NEXT
     1 PROBLEM//)
      READ(5,1215) IDUM
      IF(IDUM.GT.0) STOP 77
      READ(5,28) INPD,INPK
      IF(INPK.EQ.1) GO TO 1216
      IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216
C     NEXT RUN NOT SOLVABLE WITH PRESENT STARTING GAINS = SO STOP
      WRITE(9,3008)
      STOP 11
C     SET P = Q
  874 DO 877 I=1,NX
      DO 877 J=1,NX
  877 P(I,J)=Q(I,J)
      IF(ITERC.GT.0) GO TO 3057
C     ON SECOND ITERATION - SOLVE FOR DIFFERENCE  P= P - PI
      DO 3058 I=1,NX
      DO 3058 J=1,NX
 3058 P(I,J)=P(I,J)-PI(I,J)
 3057 CONTINUE
      ITERC=ITERC+1
C     UPDATE A AND Q FOR NEXT ITERATION ,WHERE
C       A = AN - E*(P+PI)   - (P+PI) IS TOTAL RICCATI MATRIX
C       Q = -P*E*P
C       TO SOLVE FOR THE DIFFERENCE P FROM
C
C           0 = A**P + P*A + Q
C
      DO 3050 I=1,NX
      DO 3050 J=1,NX
      EP(I,J)=0.
      A(I,J)=AN(I,J)
      DO 3050 K=1,NX
      EP(I,J)=EP(I,J)+E(I,K)*P(K,J)
 3050 A(I,J)=A(I,J)-E(I,K)*(PI(K,J)+P(K,J))
      DO 3052 I=1,NX
      DO 3052 J=I,NX
      Q(I,J)=0.
      DO 3051 K=1,NX
 3051 Q(I,J)=Q(I,J)-P(I,K)*EP(K,J)
 3052 Q(J,I)=Q(I,J)
C     BEFORE GOING TO THE NEXT ITERATION, CHECK FOR CONVERGENCE
```

Figure 84. Program DIAK Program Listing (Continued)

```
C       CONVERGENCE IS WHEN THE ABSOLUTE CHANGE IN THE ELEMENTS OF THE RI
C       MATRIX BETWEEN ITERATIONS IS LESS THAN THE ABSOLUTE VALUE OF THE
C       TIMES EE
C       ONLY CHECK THE UPPER TRIANGULAR ELEMENTS
      ICT=0
      DO 105 I=1,NX
      DO 105 J=I,NX
      API=ABS(P(I,J))
      IF(API.LT.1.E-20) GO TO 105
C    IF THE ELEMENTS ARE SMALL, CONSIDER THEM AS ZERO AND COUNT THEM AS C
      IF(API.LT.1.E+20) GO TO 888
C    IF THE ELEMENTS ARE LARGE, CONSIDER THEM AS DIVERGING AND GO TO NEXT
      WRITE(9,39)
      READ(5,1215) IDUM
      IF(IDUM.GT.0) STOP 77
      READ(5,28) INPD,INPK
      IF(INPK.EQ.1) GO TO 1216
      IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216
C    NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP
      WRITE(9,3008)
      STOP 11
888   API=ABS(PI(I,J))
      IF(API.LT.1.E-20)  GO TO 105
  106 RAT=P(I,J)/PI(I,J)
      RAT=ABS(RAT)
      IF(RAT-EE)105,105,107
C    COUNT CONVERGED ELEMENTS
  105 ICT=ICT+1
  107 CONTINUE
C    IF ICT DOES NOT EQUAL NC, THE NUMBER OF ELEMENTS, AND THE NUMBER OF
C       TIONS DOES NOT EQUAL ITER, GO TO NEXT ITERATION
  108 IF(NC-ICT)109,122,109
  109 IF(ITERC-ITER)1000,1001,1001
C       IF ITERC EQUALS ITER, NO CONVERGENCE - PRINT LAST TWO RICCATI MAT
C       AND GO TO NEXT RUN
 1001 WRITE(9,120)ITER,ICT
  120 FORMAT(1H1/7X,18H NOT CONVERGED IN I3,34H ITERATIONS-FIRST TERM TO
     1FAIL WAS I4/)
      ITERM=ITER-1
      DO 3054 I=1,NX
      DO 3054 J=1,NX
 3054 P(I,J)=P(I,J)+PI(I,J)
      WRITE(9,121)ITER
  121 FORMAT(///23H P MATRIX AT ITERATION I3//)
      CALL MP(MX,MX,NX,NX,P)
      WRITE(9,121)ITERM
      CALL MP(MX,MX,NX,NX,PI)
      WRITE(9,39)
C *** MODIFICATIONS
CR    READ(5,1215) IDUM
CR    IF(IDUM.GT.0) STOP 77
CR    READ(5,28) INPD,INPK
CR    IF(INPK.EQ.1) GO TO 1216
CR    IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216
C   NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP
CR    WRITE(9,3008)
CR    STOP 11
C *** MODIFICATIONS
  122 CONTINUE
C   COMPUTE OPTIMAL GAINS
C       K = -(D#*Q*D):*D#*Q*H - (D#*Q*D):*G1*P
      DO 3056 I=1,NX
      DO 3056 J=1,NX
 3056 P(I,J)=P(I,J)+PI(I,J)
      DO 125 I=1,NU
      DO 125 J=1,NX
```

Figure 84. Program DIAK Program Listing (Continued)

```
      AK(I,J)=-W(I,J)
      DO 125 K=1,NX
  125 AK(I,J)=AK(I,J)-W1(I,K)*P(K,J)
C   SET COMMAND FEEDFORWARD GAINS TO ZERO
      NXMNC=NX-NCS+1
      DO 86 I=1,NU
      II=I+NF-NU
      DO 86 J=NXMNC,NX
C *** MODIFICATIONS
C     A(II,J)=0.
C *** MODIFICATIONS
   86 AK(I,J)=0.
C *** MODIFICATIONS
C     RECOMPUTE A - CLOSED LOOP STABILITY MATRIX
      REWIND 2
      READ(2) A
      DO 88 I=1,NX
      DO 88 J=1,NX
      DO 88 K=1,NU
   88 A(I,J)=A(I,J)+G1(I,K)*AK(K,J)
C *** MODIFICATIONS
C   PRINT GAINS MATRIX AND RICCATI MATRIX
 4004 FORMAT(1H1/7X,13H GAINS MATRIX//)
 4010 WRITE(9,4005)
 4005 FORMAT(1H1/7X,15H RICCATI MATRIX//)
      CALL MP(MX,MX,NX,NX,P)
      WRITE(9,4004)
      CALL MP(MU,MX,NU,NX,AK)
C   RE-READ H AND M MATRICES FROM DISC
      REWIND 2
      READ(2)
      READ(2) H,AM
      IF (NCONT.LT.2) GO TO 82
C  RECOMPUTE QUADRATICS WEIGHTS ON CONTROL RATES
      NSCSS=NF-NU+1
      DO 80 I=NSCSS,NF
      II=I+NSCRR-NSCSS
      IJ=I-NSCSS+1
      DO 80 J=NSCSS,NF
      JJ=J+NSCRR-NSCSS
   80 QQ(II,JJ)=-P(I,J)*G1(I,IJ)/(H(JJ,J)*D(II,IJ))
      REWIND 2
      READ(2) F
      NUU=NU*NU
      NCU=0
      DO 84 I=1,NU
      DO 84 J=NSCSS,NF
      IF(ABS(AK(I,J)).GT..05) GO TO 84
      NCU=NCU+1
   84 CONTINUE
      IF(NCU.EQ.NUU) GO TO 85
      IF(NQ.GT.ITERQ) GO TO 85
      NQ=NQ+1
      INPK=3
      INPD=2
      GO TO 81
   85 CONTINUE
      WRITE(6,83) IRUN
   83 FORMAT(17HQ MATRIX FOR CASE,I3)
      CALL OUTP(MR,MR,NR,NR,QQ,6)
   82 CONTINUE
      WRITE(6,7776)
 7776 FORMAT(20(4H    ))
C   PUNCH IDENTIFICATION
      WRITE(6,9010) IRUN
 9010 FORMAT(21HGAINS MATRIX FOR CASE,I3)
```

Figure 84. Program DIAK Program Listing (Continued)

```
C     PUNCH OPTIMAL GAINS
         CALL OUTP(MU,MX,NU,NX,AK,6)
         WRITE(6,7776)
C     INVERT M MATRIX (IN P) FOR COMPUTATION OF KSTAR = K*M:
         DO 892 I=1,NX
         DO 892 J=1,NX
  892 P(I,J)=AM(I,J)
         CALL TDINVR(ISOL,IDSOL,NX,NX,P,MX,KWA,DET)
         IF((ISOL+IDSOL)-2) 889,889,890
C     IF M MATRIX DOESN#T INVERT, FORGET COMPUTATION OF KSTAR - SKIP TO
C        RESPONSE CALCULATIONS (STATEMENT 894)
  890 WRITE(9,40)
   40 FORMAT(1H1/7X,32H M MATRIX INVERSE DOES NOT EXIST//7X,10H IGNORE I
     1T//)
         GO TO 894
C     COMPUTE KSTAR (IN AN)
  889 DO 1280 I=1,NU
         DO 1280 J=1,NX
         AN(I,J)=0.
         DO 1280 K=1,NX
 1280 AN(I,J)=AN(I,J)+AK(I,K)*P(K,J)
C     STORE KSTAR IN WI
         DO 58 I=1,NU
         DO 58 J=1,NX
   58 WI(I,J)=AN(I,J)
C     PRINT AND PUNCH KSTAR
         WRITE(9,1281)
 1281 FORMAT(1H1/7X,13H KSTAR MATRIX/)
         CALL MP(MX,MX,NU,NX,AN)
         WRITE(6,9011) IRUN
 9011 FORMAT(,21HKSTAR MATRIX FOR CASE,I3)
         CALL OUTP(MX,MX,NU,NX,AN,6)
         WRITE(6,7776)
C     GO TO RESPONSE CALCULATIONS
         GO TO 894
  893 DO 898 I=1,NU
         DO 898 J=1,NX
         WI(I,J)=BK(I,J)
         AK(I,J)=0.
         DO 898 K=1,NX
  898 AK(I,J)=AK(I,J)+BK(I,K)*AM(K,J)
         DO 895 I=1,NX
         DO 895 J=1,NX
         A(I,J)=F(I,J)
         DO 895 K=1,NU
  895 A(I,J)=A(I,J)+G1(I,K)*AK(K,J)
         WRITE(9,42)
   42 FORMAT(1H1/7X,41H AIRCRAFT RESPONSES WITH PRESCRIBED GAINS//)
  894 DO 4052 I=1,NR
         DO 4052 J=1,NX
         DO 4052 K=1,NU
 4052 H(I,J)=H(I,J)+D(I,K)*AK(K,J)
         GO TO (850,851,851),NOCOV
  851 CONTINUE
         DO 6080 I=1,NR
         DO 6080 J=1,NR
 6080 QR(I,J)=0.
         AJ=0.
         KCOM=
 6076 KCOM=KCOM+1
         WRITE(9,41) KCOM
   41 FORMAT(1H1/7X,36H COVARIANCE ANALYSIS FOR DISTURBANCE,I3//)
         DO 4020 I=1,NX
         DO 4020 J=1,NX
 4020 E(I,J)=G2(I,KCOM)*G2(J,KCOM)
         DO 6075 I=1,NX
```

Figure 84. Program DIAK Program Listing (Continued)

```
      DO 6075 J=1,NX
 6075 P(I,J)=A(I,J)
      CALL CALI(P,E,O,KWA,NX,MX,IMAX,2,IERR,EE)
      IF(IERR.EQ.0) GO TO 896
      WRITE(9,43)
   43 FORMAT(1H1/7X,28H COVARIANCE MATRIX UNDEFINED//7X,27H IGNORE COVAR
     1IANCE ANALYSIS//)
      GO TO 855
  896 WRITE(9,4051)
 4051 FORMAT(/7X,18H COVARIANCE MATRIX//)
      CALL MP(MX,MX,NX,NX,E)
      DO 4053 I=1,NR
      DO 4053 J=1,NX
      AN(I,J)=0.
      DO 4053 K=1,NX
 4053 AN(I,J)=AN(I,J)+H(I,K)*E(K,J)
      DO 4054 I=1,NR
      DO 4054 J=1,NR
      WR(I,J)=0.
      DO 4054 K=1,NX
 4054 WR(I,J)=WR(I,J)+AN(I,K)*H(J,K)
      WRITE(9,4055)
 4055 FORMAT(1H1/7X,27H RESPONSE COVARIANCE MATRIX//)
      CALL MP(MR,MR,NR,NR,WR)
      DO 6077 I=1,NR
      DO 6077 J=1,NR
      QR(I,J)=QR(I,J)+WR(I,J)
 6077 AJ=AJ+WR(I,J)*QQ(I,J)
      DO 7015 I=1,NX
      DO 7015 J=1,NX
      P(I,J)=0.
      DO 7015 K=1,NX
 7015 P(I,J)=P(I,J)+E(I,K)*AM(J,K)
      DO 7016 I=1,NX
      DO 7016 J=1,NX
      Q(I,J)=0.
      DO 7016 K=1,NX
 7016 Q(I,J)=Q(I,J)+AM(I,K)*P(K,J)
      WRITE(9,44)
   44 FORMAT(1H1/7X,30H MEASUREMENT COVARIANCE MATRIX//)
      CALL MP(MX,MX,NX,NX,Q)
      DO 1112 I=1,NU
      DO 1112 L=1,NX
      W(I,L)=0.
      DO 1112 K=1,NX
 1112 W(I,L)=W(I,L)+AK(I,K)*E(K,L)
      DO 6085 I=1,NU
      DO 6085 J=1,NU
      DQD(I,J)=0.
      DO 6085 K=1,NX
 6085 DQD(I,J)=DQD(I,J)+W(I,K)*AK(J,K)
      WRITE(9,45)
   45 FORMAT(1H1/7X,26H CONTROL COVARIANCE MATRIX//)
      CALL MP(MU,MU,NU,NU,DQD)
      IF(NOCOV.GT.2) GO TO 2
      DO 1111 I=1,NX
      DO 1111 J=1,NX
      P(I,J)=0.
      IF(E(I,I).LT.1.E-20) GO TO 1111
      IF(E(J,J).LT.1.E-20) GO TO 1111
      P(I,J)=E(I,J)/SQRT(E(I,I)*E(J,J))
 1111 CONTINUE
      WRITE(9,46)
   46 FORMAT(1H1/7X,31H STATE CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NX,NX,P)
      DO 1113 I=1,NU
```

Figure 84. Program DIAK Program Listing (Continued)

```
      DO 1113 J=1,NX
      P(I,J)=0.
      P(I,J)=W(I,J)*AK(I,J)
 1113 CONTINUE
      WRITE(9,47)
   47 FORMAT(1H1/7X,41H CONTROL-STATE ROW-SUM CORRELATION MATRIX//)
      CALL MP(MX,MX,NU,NX,P)
      DO 1114 I=1,NR
      DO 1114 J=1,NX
      P(I,J)=0.
      P(I,J)=AN(I,J)*H(I,J)
 1114 CONTINUE
      WRITE(9,48)
   48 FORMAT(1H1/7X,42H RESPONSE-STATE ROW-SUM CORRELATION MATRIX//)
      CALL MP(MX,MX,NR,NX,P)
      DO 1115 I=1,NX
      DO 1115 J=1,NX
      P(I,J)=0.
      IF(Q(I,I).LT.1.E-20) GO TO 1115
      IF(Q(J,J).LT.1.E-20) GO TO 1115
      P(I,J)=Q(I,J)/SQRT(Q(I,I)*Q(J,J))
 1115 CONTINUE
      WRITE(9,49)
   49 FORMAT(1H1/7X,37H MEASUREMENT CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NX,NX,P)
      DO 1300 I=1,NX
      DO 1300 J=1,NX
      P(I,J)=0.
      DO 1301 K=1,NX
1301  P(I,J)=P(I,J)+AM(I,K)*F(K,J)
      P(I,J)=P(I,J)*AM(I,J)
1300  CONTINUE
      WRITE(9,1302)
1302  FORMAT(1H1/7X,45H MEASUREMENT-STATE ROW-SUM CORRELATION MATRIX//)
      CALL MP(MX,MX,NX,NX,P)
      DO 1116 I=1,NU
      DO 1116 J=1,NX
      P(I,J)=0.
      DO 1117 K=1,NX
 1117 P(I,J)=P(I,J)+W(I,K)*AM(J,K)
      P(I,J)=P(I,J)*W1(I,J)
 1116 CONTINUE
      WRITE(9,50)
   50 FORMAT(1H1/7X,47H CONTROL-MEASUREMENT ROW-SUM CORRELATION MATRIX//
     1)
      CALL MP(MX,MX,NU,NX,P)
2     CONTINUE
      DO 63 I=1,NX
63    Q(I,I)=SQRT(Q(I,I))
      DO 64 I=1,NU
64    DQD(I,I)=SQRT(DQD(I,I))
      WRITE(9,65) ((I,DQD(I,I)),I=1,NU)
65    FORMAT(1H1//20X,16H R.M.S. CONTROLS/,(5X,I3,13X,E15.8))
      WRITE(9,66) ((I,Q(I,I)),I=1,NX)
66    FORMAT(1H1//20X,20H R.M.S. MEASUREMENTS/(5X,I3,13X,E15.8))
      DO 4056 I=1,NR
 4056 WR(I,I)=SQRT(WR(I,I))
      WRITE(9,4057) ((I,WR(I,I)),I=1,NR)
 4057 FORMAT(1H1//20X,17H R.M.S. RESPONSES/(5X,I3,13X,E15.8))
      IF(KCOM.LT.NN) GO TO 6076
      WRITE(9,60)
60    FORMAT(1H1/7X,33H TOTAL RESPONSE COVARIANCE MATRIX//)
      CALL MP(MR,MR,NR,NR,QR)
      DO 61 I=1,NR
      DO 61 J=1,NR
      P(I,J)=0.
```

Figure 84. Program DIAK Program Listing (Continued)

```
      IF(QR(I,I).LT.1.E-20) GO TO 61                                    DIAK 923
      IF(QR(J,J).LT.1.E-20) GO TO 61                                    DIAK 924
      P(I,J)=QR(I,J)/SQRT(QR(I,I)*QR(J,J))                              DIAK 925
61    CONTINUE                                                          DIAK 926
      WRITE(9,62)                                                       DIAK 927
62    FORMAT(1H1/7X,45H TOTAL RESPONSE CROSS-CORRELATION MATRIX//)      DIAK 928
      CALL  MP(MX,MX,NR,NP,P)                                           DIAK 929
      DO 6082 I=1,NP                                                    DIAK 930
 6082 QR(I,I)=SQRT(QR(I,I))                                             DIAK 931
      WRITE(9,6081) ((I,QR(I,I)),I=1,NR)                                DIAK 932
 6081 FORMAT(1H1/7X,22HTOTAL R.M.S. RESPONSES/(3X,I3,3X,E15.8))         DIAK 933
      WRITE(9,6078) AJ                                                  DIAK 934
 6078 FORMAT(1H1//20X,17HQUADRATIC COST = ,E15.8)                       DIAK 935
  850 CONTINUE                                                          DIAK 936
      IF(NOP.EQ.0) GO TO 897                                            DIAK 937
      IF(NSTEP.EQ.0.AND.NRAND.EQ.0) GO TO 897                           DIAK 938
      CALL  TIMER(A,G2,H,X,XT,DX,DX1,XLDXL,GN,GS,R,IPLR,ITITL,IUNIT,CL, DIAK 939
     1T,DT,ST,YMAX,YMIN,IFLT,IRUN,IDATE,NSTEP,NRAND,NPLOT,NPRIN,NN,NX,NFDIAK 940
     2,NG,NGS,NR,MXR,MN,MX,YPOIN,NOP,NAME1,NAME2,SCAL,NEWY,T1,T2,NGLG)  DIAK 941
  897 CONTINUE                                                          DIAK 942
      CALL  POLES(NX,A,MX,RR,RI)                                        DIAK 943
      ITERC=0                                                           DIAK 944
 1200 READ(5,1215) IDUM                                                 DIAK 945
      IF(IDUM.GT.0)GO TO 7777                                           DIAK 946
 1215 FORMAT(2I1)                                                       DIAK 947
      READ(5,28) INPQ,INPK                                              DIAK 948
 1216 IRUN=IRUN+1                                                       DIAK 949
      REWIND 2                                                          DIAK 950
      READ(2) F                                                         DIAK 951
      READ(2) H,AM                                                      DIAK 952
      READ(5,28) NCONT                                                  DIAK 953
      READ(5,28) NOCOV,NSTEP,NRAND,NPRIN,NPLOT                          DIAK 954
      WRITE(9,37) INPQ,INPK,NCONT,NOCOV,NSTEP,NRAND,NPRIN,NPLOT         DIAK 955
      IF(INPQ.EQ.1) GO TO 1210                                          DIAK 956
      WRITE(9,1271) IFLT,IRUN                                           DIAK 957
      GO TO (1210,1220,1230,1240,1250),INPQ                             DIAK 958
 7777 CONTINUE                                                          DIAK 959
      ENDFILE 6                                                         DIAK 960
      END                                                               DIAK 961
```

Figure 84. Program DIAK Program Listing (Concluded)

```
      OVERLAY(KON2,2,0)
      PROGRAM FFOC
      DIMENSION F(40,40),G1(40,06),G2(40,02),H(40,40) ,
     * D(40,06),AM(40,40),Q(40,40),AK(06,40),DK(06,40),DJDK(06,40),
     * AKG(06,40),XI(09,09)  ,HK(06,40),A(40,40),X(40,40),C(40,40),
     * P(40,40),HDK(40,40),S(40,40),RR(608),DELK(06,40),AKP(06,40)
      DIMENSION U(40,40),V(9,9),E(9,9),FS(40,40),DQ(6,40) ,
     *  DQD(06,06),KWA(50),T(50,50),HB(50,40),DJV(50),DJVT(50),IF(50),
     * JF(50),AMT(50,40),Y(40,40),Z(40,40),NORD(40)
C
C   ARRAY DIMENSIONS
C
      MX=40
      MR=40
      MNN=2
      MU=6
      MM=40
      MFF=9
      MFB=4
      MF=50
C   INPUT INTEGER PARAMETERS
      READ(5,1) IMAX,NITM,NOPR,NOCOV,NBEGIN  ,NDIAK
      READ(5,1) NX,NR,NU,NN,NFF,NF
1     FORMAT(40I2)
      READ(5,1) (NORD(I),I=1,NX)
      ITER=
      NM=NX
      WRITE(9,9) IMAX,NITM
9     FORMAT(1H1/7X,39H MAXIMUM NO. OF INNER LOOP ITERATIONS =,I3//7X,
     139H MAXIMUM NO. OF OUTER LOOP ITERATIONS =,I3//)
      WRITE(9,11) NOCOV,NBEGIN,NOPR
11    FORMAT(//7X,8H NOCOV =,I3,5X,9H NBEGIN =,I3,5X,7H NOPR =,I3//)
      WRITE(9,13) NX,NR,NU,NN,NFF,NF
13    FORMAT(//7X,16H NO. OF STATES =,I3,5X,19H NO. OF RESPONSES =,I3//
     17X,18H NO. OF CONTROLS =,I3,5X,22H NO. OF DISTURBANCES =,I3//
     27X,29H NO. OF FEEDFORWARD  STATES =,I3//7X,26H NO. OF FIXED-FORM G
     3AINS =,I3//)
C
C  INITIAL STEP SIZE
   60 READ(5,1277) EPSI
C
C   INPUT REAL PARAMETERS
      READ(5,1277) AJSTAR
      AJT=AJSTAR
      WRITE(9,5035)
5035  FORMAT(  //7X,28HLOWEST COST EXPECTED(AJSTAR)/)
      WRITE(9,5033) AJSTAR
5033  FORMAT(//7X,(6G10.4))
      WRITE(9,2) (NORD(I),I=1,NX)
2     FORMAT(//7X,27H STATES ARE ORDERED AS SUCH//(7X,30I3))
      READ(5,1277) DROC
      READ(5,1277) ALAM,DELT,ALAMD
 1277 FORMAT(6G10.4)
    3 NG=1
      NC=1
      NUNST=0
C
C   ROWS AND COLUMNS OF FIXED GAINS -- K1
      READ(5,1) (IF(I),JF(I),I=1,NF)
      WRITE(9,20)
20    FORMAT(//7X,29HFIXED GAINS       ROW   COLUMN//)
      WRITE(9,201) (IF(I),JF(I),I=1,NF)
201   FORMAT(21X,2I6)
```

Figure 85. Program FFOC Program Listing

```
C
C  INPUT SYSTEM MATRICES -- F,G1,G2,H,D,M,Q
      CALL ZERO (F,MX,MX)
      CALL ZERO(G1,MX,MU)
      CALL ZERO(G2,MX,MNN)
      CALL ZERO(H,MR,MX)
      CALL ZERO(D,MR,MU)
      CALL ZERO(AM,MM,MX)
      CALL ZERO (Q,MR,MR)
      CALL INPT(F,MX,MX)
      CALL INPT(G1,MX,MU)
      CALL INPT(G2,MX,MNN)
      CALL INPT(H,MR,MX)
      CALL INPT(D,MR,MU)
      CALL INPT(AM,MM,MX)
      CALL INPT (Q,MR,MR)
C  INPUT OPTIMAL GAINS
      CALL ZERO(AKG,MU,MX)
      CALL INPT(AKG,MU,MX)
      CALL SHUF(F,G1,G2,H,AM,AKG,Y,NORD,MX,NX,MR,NR,MM,NM,MU,NU,MNN,NN)
      WRITE(9,1010)
 1010 FORMAT(1H1/7X,10H F  MATRIX//)
      CALL MP(MX,MX,NX,NX,F)
      WRITE(9,1011)
 1011 FORMAT(1H1/7X,10H G1 MATRIX//)
      CALL MP(MX,MN,NX,NU,G1)
      WRITE(9,1012)
 1012 FORMAT(1H1/7X,10H G2 MATRIX//)
      CALL MP(MX,MNN,NX,NN,G2)
      WRITE(9,1013)
 1013 FORMAT(1H1/7X,10H H  MATRIX//)
      CALL MP(MR,MX,NR,NX,H)
      WRITE(9,1014)
 1014 FORMAT(1H1/7X,10H D  MATRIX//)
      CALL MP(MR,MN,NR,NU,D)
      WRITE(9,1015)
 1015 FORMAT(1H1/7X,10H M  MATRIX//)
      CALL MP(MM,MX,NM,NX,AM)
      WRITE(9,1016)
 1016 FORMAT(1H1/7X,10H Q  MATRIX//)
      CALL MP(MR,MR,NR,NR,Q)
      CALL ZERO(AKP,MU,MM)
      L=1
      DO 1750 I=1,NU
      DO 1750 J=1,NM
      IF(L.GT.NF) GO TO 1750
      IF(I.NE.IF(L)) GO TO 1750
      IF(J.NE.JF(L)) GO TO 1750
      AKP(I,J)=1.
      DO 1751 N=1,NX
1751  AMT(L,N)=AM(J,N)
      L=L+1
1750  CONTINUE
      WRITE(9,1017)
 1017 FORMAT(1H1/7X,40H MEASUREMENT MATRIX FOR FIXED FORM GAINS//)
      CALL MP(MF,MX,NF,NX,AMT)
      WRITE(9,1500)
1500  FORMAT(1H1/7X,22H OPTIMAL RICCATI GAINS//)
      CALL MP(MU,MX,NU,NM,AKG)
      DO 1501 I=1,NM
      DO 1501 J=1,NX
1501  S(I,J)=AM(I,J)
      CALL TDINVR(ISOL,IDSOL,NX,NX,S,MX,KWA,DET)
      IF((ISOL+IDSOL).LE.2) GO TO 1502
      WRITE(9,1403) ISOL,IDSOL
1403  FORMAT(//7X,51H MEASUREMENT MATRIX IS NOT INVERTIBLE, ERROR CODE =
```

Figure 85. Program FFOC Program Listing (Continued)

```
     1,213//)
      STOP 33
1502  DO 1405 I=1,NU
      DO 1405 J=1,NX
      AK(I,J)=0.
      DO 1405 K=1,NX
1405  AK(I,J)=AK(I,J)+AKG(I,K)*S(K,J)
C
C     DEFINE K2
      CALL INPT(AK,MU,MM)
      DO 1404 I=1,NU
      DO 1404 J=1,NX
      BK(I,J)=AK(I,J)*(1.-AKP(I,J))
1404  AK(I,J)=AK(I,J)*AKP(I,J)
      CALL INPT(BK,MU,MM)
      WRITE(9,1018)
 1018 FORMAT(1H1/7X,23H INITIAL GAINS -- K1(1)//)
      CALL MP(MU,MX,NU,NM,AK )
      WRITE(9,1019)
 1019 FORMAT(//7X,10H K2 MATRIX//)
      CALL MP(MU,MM,NU,NM,BK)
      IF(ALAM.LT..99) GO TO 1406
      DO 1407 I=1,NU
      DO 1407 J=1,NM
1407  DELK(I,J)=0.
      GO TO 1408
C
C    INPUT PRESENT FIXED GAINS -- K1(LAMBDA)
1406  CALL ZERO(AK,MU,MM)
      CALL INPT(AK,MU,MM)
      WRITE(9,1020)
 1020 FORMAT(//7X,34H PRESENT FIXED GAINS -- K1(LAMBDA)//)
      CALL MP(MU,MM,NU,NM,AK)
C
C    INPUT FIXED PREDICTOR -- DELK1(LAMBDA)
      CALL ZERO(DELK,MU,MM)
      CALL INPT(DELK,MU,MM)
1408  WRITE(9,1021)
 1021 FORMAT(//7X,35H PRESENT PREDICTOR -- DELK1(LAMBDA)//)
      CALL MP(MU,MM,NU,NM,DELK)
C
C    TAKE STEP IN LAMBDA
172   ALAM=ALAM-DELT
      WRITE(9,173) ALAM
173   FORMAT(//7X,9HLAMBDA = ,F9.3)
C
C    PREDICT GAINS FOR NEW LAMBDA
      DO 310 I=1,NU
      DO 310 J=1,NM
      DK(I,J)=DELK(I,J)
      DELK(I,J)=AK(I,J)
  310 AK(I,J)=AK(I,J)+DK(I,J)
C
C  INITIAL CONDITIONS
C
      NL=1
      LAST=2
      NCHK=
      NGD=0
      NIT=0
      EPS=EPSI
      AJI=1.*AJT
7     AJL=AJT
      WRITE(9,4051) NIT
 4051 FORMAT( ///7X,9HITERATION,I3)
      WRITE(9,59) EPS
```

Figure 85. Program FFOC Program Listing (Continued)

```
C
C    PRINT GAINS
      WRITE(9,4004)
4004  FORMAT(//7X,13H GAINS MATRIX//)
      CALL  MP(MU,MM,NU,NM,AK)
C
C    INITIALIZE GRADIENT PROJECTION
      IF(NG ,NE.3) GO TO 5
      DO 16 I=1,NU
      DO 16 J=1,NM
16    AKP(I,J)=AK(I,J)
C
C   INITIALIZE ARRAYS
C
C   COMPUTE F+G1*K*(LAMBDA)*M=A
C
    5 DO 12 J=1,NX
      DO 12 I=1,NU
      C(I,J)=0.
      DO 12 K=1,NM
      C(I,J)=C(I,J)+(AK(I,K)+BK(I,K)*ALAM)*AM(K,J)
   12 CONTINUE
      IF(LAST.NE.1) GO TO 66
      DO 151 I=1,NU
      DO 151 J=1,NM
  151 DJDK(I,J)=AK(I,J)+ALAM*BK(I,J)
      WRITE(9,1022)
 1022 FORMAT(//7X,37H K*(LAMBDA) FOR RESPONSE CALCULATIONS//)
      CALL MP(MU,MM,NU,NM,DJDK)
   66 CONTINUE
      DO 8 I=1,NX
      DO 8 J=1,NX
      A(I,J)=F(I,J)
      DO 8 K=1,NU
    8 A(I,J)=A(I,J)+G1(I,K)*C(K,J)
      IF(NL.NE.1) GO TO 184
C
C    CHECK FOR STABILITY OF A
      CALL POLES(NX,A,MX,RR,M)
      KK=0
      II=1
921   IF(RR(II).LT.0.) GO TO 185
C
C    IF UNSTABLE -- HALVE DELTA LAMBDA AND PREDICTOR
   69 I=NUNST+1
      WRITE(9,188) I
  188 FORMAT(//7X,24HUNSTABLE -- CHANGE GAINS//7X,I3,14HTH INSTABILITY/)
      IF(NUNST.EQ.3) GO TO 5071
      ALAM=ALAM+DELT
      IF(NUNST.EQ.1) GO TO 5071
C
C   FIRST OR THIRD INSTABILITY -- HALVE PREDICTOR
      DO 5040 I=1,NU
      DO 5040 J=1,NM
      AK(I,J)=AK(I,J)-DK(I,J)
5040  DELK(I,J)=DK(I,J)/2.
      AJT=AJSTAR
      NUNST=NUNST+1
      GO TO 172
C
C   SECOND OR FOURTH INSTABILITY -- HALVE DELTA LAMBDA
 5071 DELT=DELT*.5
C *** MODIFICATIONS
      IF(DELT.LE.1.0E-06)WRITE(9,7740)DELT
 7740 FORMAT(1H1,//,1X,51H*** EXIT ON DETECTING VERY SMALL VALUE FOR DEL
     1T ***,G10.4)
```

Figure 85. Program FFOC Program Listing (Continued)

```
      IF(DELT.LE.1.0E-06)GO TO 1730
C *** MODIFICATIONS
      DO 5072 I=1,NU
      DO 5072 J=1,NM
      DELK(I,J)=DK(I,J)
 5072 AK(I,J)=AK(I,J)-DK(I,J)
C
C  STOP AND PUNCH GAINS AND PREDICTOR ON FOURTH INSTABILITY
      IF(NUNST.EQ.3) GO TO 5041
C *** MODIFICATIONS
C     ALAMD=ALAMD+DELT
C *** MODIFICATIONS
      AJT=AJSTAR
      NUNST=2
      GO TO 172
185   IF(RR(II+1).EQ.0.   ) GO TO 922
      KK=KK+2
      GO TO 923
922   KK=KK+1
923   IF(KK.EQ.N ) GO TO 924
      II=II+2
      GO TO 921
924   CONTINUE
C
C   RECOMPUTE A
      DO 187 I=1,NX
      DO 187 J=1,NX
      A(I,J)=F(I,J)
      DO 187 K=1,NU
187   A(I,J)=A(I,J)+G1(I,K)*C(K,J)
C
C  COMPUTE  H+DKM, R=(H+DKM)Q(H+DKM), DQ
  184 DO 7012 I=1,NR
      DO 7012 J=1,NX
      HDK(I,J)=H(I,J)
      DO 7012 K=1,NU
 7012 HDK(I,J)=HDK(I,J)+D(I,K)*C(K,J)
      IF(LAST.EQ.1) GO TO 120
      DO 7013 I=1,NR
      DO 7013 J=1,NX
      C(I,J)=0.
      DO 7013 K=1,NR
 7013 C(I,J)=C(I,J)+Q(I,K)*HDK(K,J)
      DO 7014 I=1,NX
      DO 7014 J=1,NX
      R(I,J)=0.
      DO 7014 K=1,NR
 7014 R(I,J)=R(I,J)+HDK(K,I)*C(K,J)
      DO 18 I=1,NU
      DO 18 J=1,NR
      DQ(I,J)=0.
      DO 18 K=1,NR
   18 DQ(I,J)=DQ(I,J)+D(K,I)*Q(K,J)
      DO 434 I=1,NU
      DO 434 J=1,NU
      DQD(I,J)=0.
      DO 434 K=1,NR
  434 DQD(I,J)=DQD(I,J)+DQ(I,K)*D(K,J)
C
C   COMPUTE COVARIANCE MATRIX
      CALL COVAR(XI,A,C,X,G2,S,E,ES,V,U,NX,NFF,NN,MX,MFF,MFB,MNN,IMAX,
     1ITER,1,IERR,KWA)
      IF(IERR.EQ.0) GO TO 17
      IF(NL.NE.1) GO TO 4
      DO 6 I=1,NU
      DO 6 J=1,NX
```

Figure 85. Program FFOC Program Listing (Continued)

```
      C(I,J)=0.
      DO 6 K=1,NM
6     C(I,J)=C(I,J)+(AK(I,K)+BK(I,K)*ALAM)*AM(K,J)
      GO TO 69
4     CONTINUE
      AJT=1.E+20
      IF(NG.NE.1) GO TO 43
      GO TO 1756
   17 ITER=1
C
C  CALCULATE COST
C
      AJT=0.
      DO 34 I=1,NX
      DO 34 J=1,NX
   34 AJT=AJT+R(I,J)*X(I,J)
      WRITE(9,122) AJT
  122 FORMAT(//7X, 8H COST = ,E15.8)
      IF(NBEGIN.EQ.0) GO TO 10
      IF(AJT.LT.0.) GO TO 1510
      IF(AJT.LT.AJI) GO TO 10
1510  WRITE(9,1402)
1402  FORMAT(//7X,43H COST EXCEEDS 10 TIMES LOWEST COST EXPECTED//)
      WRITE(9,14)
14    FORMAT(1H1/7X,18H COVARIANCE MATRIX//)
      CALL MP(MX,MX,NX,NY,X)
      STOP 11
   10 IF(NG.NE.1) GO TO 850
130   IF(NG1.EQ.2) GO TO 67
C
C   COMPUTE XM, MXM INVERSE
      DO 125 I=1,NX
      DO 125 J=1,NM
      C(I,J)=0.
      DO 125 K=1,NX
  125 C(I,J)=C(I,J)+X(I,K)*AM(J,K)
      CALL TRANS(AMT,X,T,DQD,NX,MX,NF,MU,MF,RR,IF)
C
C  COSTATE CALCULATION
C
      CALL COSTAT (R,A,S,X,ES,Y,Z,E,U,V,KWA,MX,MFR,MFF,NX,NFF,IMAX,IERR)
      IF(IERR.EQ.0) GO TO 1400
      WRITE(9,1401)
1401  FORMAT(//7X,45H COSTATE MATRIX UNSTABLE WHEN STATE MATRIX IS//)
      STOP 11
C
C  GRADIENT CALCULATION
C
C
C  COMPUTE 2DQ(H+DKM)XM(MXM)-1 -- 2G1*SXM(MXM)-1
1400  DO 19 I=1,NU
      DO 19 J=1,NX
      R(I,J)=0.
      DO 19 K=1,NR
   19 R(I,J)=R(I,J)+DQ(I,K)*HDK(K,J)
      DO 30 I=1,NU
      DO 30 J=1,NX
      X(I,J)=0.
      DO 30 K=1,NX
   30 X(I,J)=X(I,J)+G1(K,I)*S(K,J)
      DO 126 I=1,NU
      DO 126 J=1,NM
      DJDK(I,J)=0.
      DO 126 K=1,NX
  126 DJDK(I,J)=DJDK(I,J)+(R(I,K)+X(I,K))*C(K,J)*2.
      GO TO 850
```

Figure 85. Program FFOC Program Listing (Continued)

```
67    CONTINUE
C
C   PROJECTED GRADIENT
      DO 106 I=1,NU
      DO 106 J=1,NM
106   DJDK(I,J)=AKP(I,J)-AK(I,J)
      GO TO 850
C
C  CALCULATE RESPONSES
C
120   CONTINUE
      GO TO (853,851,851),NOCOV
  851 CALL RESP(C,A,G2,AM,DJDK,X,Y,Z,S,R,ES,E,U,V,X1,DQ,AKG,DQD,HDK,KWA,
     1NX,NFF,NN,NM,NU,NR,MX,MFF,MFB,MNN,MM,MU,MR,ITER,IMAX,IERR,NOCOV)
      ITER=
  853 CALL POLES(NX,A,MX,RR,M)
  850 CONTINUE
      IF(LAST.EQ.1) GO TO 57
      IF(NL.EQ.1) AJLAT=AJT
      NBEGIN = 0
      IF(NG.NE.1) GO TO 43
      IF (NGD.EQ.2) GO TO 1756
      CALL TDINVR(ISOL,IDSOL,NF,NF,T,MF,KWA,DET)
      IF((ISOL+IDSOL).LE.2) GO TO 852
      WRITE(9,1023)
 1023 FORMAT(//7X,39H GRADIENT TRANSFORMATION NOT INVERTABLE//)
      STOP 22
  852 WRITE(9,1024)
 1024 FORMAT(  //7X,31H GRADIENT TRANSFORMATION MATRIX//)
      CALL  INSCR(T,DJDK,DJV,DJVT,IF,JF,NF,NJ,NM,MU,MM,MF)
      CALL  POLES(NF,T,MF,RR,M)
1756  IF(NL.GT.0) GO TO 500
      IF(NGBB.GT.0) GO TO 500
C
C    CORRECT FOR INSTABILITY WHILE COMPUTING GRADIENT
     NGB=1
      IF(AJT.GT.AJM) NGB=0
      IF(AJT.LT.0.) NGB=0
      IF(NGB.NE.0) GO TO 500
      NGBB=1
      EPS=EPSO
      DO 501 I=1,NU
      DO 501 J=1,NM
      AK(I,J)=AK(I,J)-DK(I,J)
      DK(I,J)=EPSO*DK(I,J)/EPSS
      IF(NCO.EQ.2) AK(I,J)=AK(I,J)+2.*DK(I,J)
      IF(NCO.EQ.3) AK(I,J)=AK(I,J)+.5*DK(I,J)
      IF(NCO.EQ.5) AK(I,J)=AK(I,J)+DK(I,J)
501   CONTINUE
      GO TO 7
500   IF(NCHK.EQ.0) GO TO 5006
C
C    COMPUTE  RATIO OF COSTS
      ROC=AJT/AJLAT
      WRITE(9,5030) ROC
5030  FORMAT(//,7X,17HRATIO OF COSTS = ,F10.4)
      IF(ROC.GT.DROC) LAST=1
      AJLAT=AJT
      IF(NIT.GE.NITM) LAST=1
C
C  NORMALIZE GRADIENT  AND COMPUTE DELTA GAINS
C
5006  SUM=0.
5008  DO 40 I=1,NU
      DO 40 J=1,NM
   40 SUM=SUM+DJDK(I,J)*DJDK(I,J)
```

Figure 85. Program FFOC Program Listing (Continued)

```
      SUM=SQRT(SUM)
      WRITE(9,5031) SUM
5031  FORMAT(//,7X,16HGRADIENT NORM = ,F15.8)
5009  DO 103 I=1,NU
      DO 103 J=1,NM
103   DJDK(I,J)=DJDK(I,J)/SUM
121   WRITE(9,39)
39    FORMAT(//7X,19HNORMALIZED GRADIENT//)
38    CALL  IP(MU,MM,NU,NM,DJDK)
5010  NL=0
      IF(LAST.EQ.1) GO TO 5
C
C     COUNT  GRADIENT DIRECTIONS
      NGBB=
      NGD=NGD+1
      NCHK=
      IF(NOPR.GT.0) GO TO 102
      IF(NGD.EQ.3) NCHK=1
      IF(NGD.EQ.3) NGD=0
      GO TO 110
  102 IF(NGD.EQ.2) NCHK=1
      IF(NGD.EQ.2) NGD=0
  110 AJOG=AJT
      AJOGL=AJL
5011  DO 42 I=1,NU
      DO 42 J=1,NM
      AKG(I,J)=AK(I,J)
   42 DK(I,J)=-EPS             *DJDK(I,J)
      NG=0
      NC=1
      NCO=0
      GO TO 44
C
C  STEP SIZE LOGIC
C
43    IF(AJT.LT.0.) GO TO 301
      IF(AJT.LT.AJI) GO TO 41
C
C   UNSTABLE -- HALVE  STEP SIZE
301   NCO=NC
      NC=1
      NIT=NIT-1
      IF(NCO.GT.1) NIT=NIT-1
      AJT=AJOG
      AJL=AJOGL
      EPS=EPS/2.
C *** MODIFICATIONS
      IF(EPS.LE.1.0E-06)WRITE(9,7720)EPS
 7720 FORMAT(1H1,//,1X,50H*** EXIT ON DETECTING VERY SMALL VALUE FOR EPS
     1 ***,G12.4)
      IF(EPS.LE.1.0E-06)GO TO 1735
C *** MODIFICATIONS
5012  DO 123 I=1,NU
      DO 123 J=1,NM
      AK(I,J)=AKG(I,J)
123   DK(I,J)=.5*DK(I,J)
      GO TO 44
41    IF(NC.GT.1) GO TO 45
      IF(AJT.GT.AJL) GO TO 47
      IF(NCO.EQ.1) GO TO 47
C
C   DOUBLE STEP SIZE
5013  DO 46 I=1,NU
      DO 46 J=1,NM
46    AK(I,J)=AK(I,J)+DK(I,J)
      NIT=NIT+1
```

Figure 85. Program FFOC Program Listing (Continued)

```
      NC=2
      AJLL=AJL
      GO TO 7
C
C   HALVE  STEP SIZE
47    DO 48 I=1,NU
      DO 48 J=1,NM
48    AK(I,J)=AK(I,J)-.5*DK(I,J)
5014  NIT=NIT+1
      NC=3
      AJLL=AJL
      GO TO 7
C
C   COMPUTE NEW STEP SIZE
45    IF(NC.EQ.3) GO TO 49
      AJD=AJL-AJT
      IF(AJD.LT.0.) NC=5
      AJDD=AJLL-AJL
      IF(AJD.LT.AJDD) GO TO 431
      EPSS=2.*EPS
      GO TO 432
431   B=(AJT-AJL*2.+AJLL)/(2.*EPS*EPS)
      GO TO 50
49    AJD=AJT-AJL
      AJDD=AJLL-AJT
      IF(AJDD.LT.0.) NC=4
      IF(AJD.LT.0.) GO TO 433
      IF(AJD.LT.AJDD) GO TO 433
      EPSS=EPS
      GO TO 432
433   B=-(4.*AJT-2.*AJL-2.*AJLL)/(EPS*EPS)
50    AB=(AJL-AJLL-EPS*EPS*B)/EPS
      EPSS=-AB/(2.*B)
432   EP=EPSS
      AJM=AMIN1(AJT,AJL,AJLL)
      EPSO=EPS
      NCO=NC
      IF(NC.EQ.3) GO TO 51
      IF(NC.EQ.4) GO TO 51
5015  DO 52 I=1,NU
      DO 52 J=1,NM
52    AK(I,J)=AK(I,J)-2.*DK(I,J)
      GO TO 53
51    DO 54 I=1,NU
      DO 54 J=1,NM
54    AK(I,J)=AK(I,J)-.5*DK(I,J)
53    NC=1
      NG=1
      DO 55 I=1,NU
      DO 55 J=1,NM
      DK(I,J)=EPSS*DK(I,J)/EPS
55    AK(I,J)=AK(I,J)+DK(I,J)
5016  NIT=NIT+1
      IF(EP.GT.0.) GO TO 175
      WRITE(9,59) EP
      EPS=-EPSS
      NCO=4
      GO TO 7
175   EPS=EPSS
      GO TO 7
   44 DO 65 I=1,NU
      DO 65 J=1,NM
   65 AK(I,J)=AK(I,J)+DK(I,J)
5017  NIT=NIT+1
      GO TO 7
57    NC=1
```

Figure 85. Program FFOC Program Listing (Continued)

```
   59 FORMAT( ///7X,12HSTEP SIZE = ,E15.8)
      NG=1
      AJSTA?=AJT
      NUNST=0
C
C   INITIALIZE NEW PREDICTOR
      DO 311 I=1,NU
      DO 311 J=1,NM
  311 DELK(I,J)=AK(I,J)-DELK(I,J)
C
C   PRINT PREDICTOR
5018  WRITE(9,313)
  313 FORMAT(///7X,14H NEW PREDICTOR//)
      CALL  'P(MU,MM,NU,NM,DELK)
      IF(ALAM.GT.ALAMD) GO TO 172
      GO TO 1730
   86 FORMAT(3G10.4,30X,16H ALAM DELT ALAMD)
C
C  STOP PROGRAM -- PRINT PREDICTOR, PUNCH OUTPUT
 5041 WRITE(9,313)
      CALL  'P(MU,MM,NU,NM,DELK)
C *** MODIFICATIONS
 1730 CONTINUE
      WRITE(1,7701)
 7701 FORMAT(9HFFOC DATA)
      WRITE(1,7702)
 7702 FORMAT(15HALAM,DELT,ALAMD)
      WRITE(1,7703)ALAM,DELT,ALAMD
 7703 FORMAT(3G10.4)
      WRITE(1,7704)
 7704 FORMAT(11HGAIN MATRIX)
      CALL  OUTP(MU,MM,NU,NM,AK,1)
      WRITE(1,7705)
 7705 FORMAT(20(4H    ))
      WRITE(1,7706)
 7706 FORMAT(15HPREDICTOR GAINS)
      CALL  OUTP(MU,MM,NU,NM,DELK,1)
      WRITE(1,7705)
      ENDFILE 1
 5025 CONTINUE
C *** MODIFICATIONS
      END
```

Figure 85. Program FFOC Program Listing (Concluded)

```
      OVERLAY(KON2,3,0)
      PROGRAM DATAK
C
C     PURPOSE - TO SET UP DIMENSIONS AND CALL DATA PREPARATION PROGRAMS
C     ANALISIS-A F KONAR/J K MAHESH-THE HONEYWELL INC.
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        DERRM
C        ERRM
C        DDIAK
C        DFFOC
C        DLSA
C        FINK
C
      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),
     1JQ,JS,JSD,JF,JD
      COMMON /INF/ NXM,NRM,NUM,CODE,MS1,MS2,MS3,MS4
      DIMENSION S1(08500)
C     DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION S2(17000)
C     DIMENSION B1(NXM,NUM),B2(NXM,NUM)
C     DIMENSION C1(NRM,NXM),C3(NRM,NAM)
C     DIMENSION D11(NRM,NUM),BK(NUM,NRM),BKC3(NUM,NXM)
C
C     DIMENSION CC(NXRM,NXRUM),NAME(NXRUM)
      DIMENSION S3(00001)
      DIMENSION S4(00001)
      DATA HDDIA,HDFFO,HDLSA/4H$DIA,4H$FFO,4H$LSA/
      DATA HPRIN/4HPRIN/
C
C     COMPUTE ARRAY START INDEXES
C
      NXRM=NXM+NRM
      NXRUM=NXRM+NUM
      M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
      M5=M4+NRM*NUM
      N1=1 $ N2=N1+NXM*NUM $ N3=N2+NXM*NUM $ N4=N3+NRM*NXM
      N5=N4+NRM*NXM $ N6=N5+NRM*NUM $ N7=N6+NUM*NRM $ N8=N7+NUM*NXM
      K1=1 $ K2=K1+NXRM*NXRUM
      K3=K2+NXRUM
C
C     CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
C
      IF(K3.GT.N8)N8=K3
      IF((M5.GT.MS1).OR.(N8.GT.MS2))
     1CALL DERRM(M5,N8,MS3,MS4,MS1,MS2,MS3,MS4,3,0,4HDATA,4HK    ,IW)
C
C     CALL DATA PREPARATION PROGRAMS
C
      IF(CODE.EQ.HDDIA)GO TO 160
      IF(CODE.EQ.HDFFO)GO TO 260
      IF(CODE.EQ.HDLSA)GO TO 360
      CALL ERRM(1,4HDATA,4HK    ,3,0,IW)
  160 CONTINUE
      CALL DDIAK(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3)
     1,S2(N4),S2(N5),S2(N6),NXM,NRM,NUM)
      GO TO 460
  260 CONTINUE
      CALL DFFOC(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3)
     1,S2(N4),S2(N5),S2(N6),NXM,NRM,NUM)
      GO TO 460
  360 CONTINUE
```

Figure 86. Program DATAK Program Listing

```
      CALL FLSA(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3)     DATAK 65
     1,S2(N4),S2(N5),S2(N6),S2(N7),NX,NP,NU,NXM,NRM,NUM)             DATAK 66
      CALL FINK(S1(M1),S1(M2),S1(M3),S1(M4),S2(K1),S2(K2),           DATAK 67
     1NX,NR,NU,NXM,NRM,NUM,NXRM,NXRUM)                               DATAK 68
  460 CONTINUE                                                       DATAK 69
C                                                                    DATAK 70
C     RETURN TO MAIN OVERLAY                                         DATAK 71
C                                                                    DATAK 72
      END                                                            DATAK 73
```

Figure 86. Program DATAK Program Listing (Concluded)

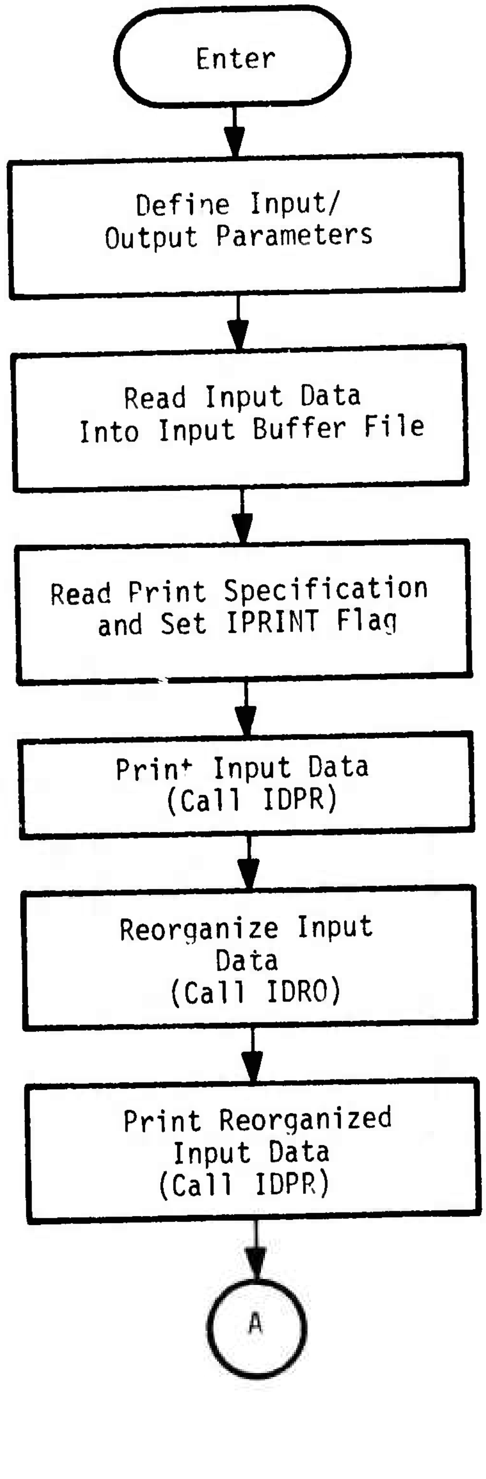

Figure 87. Subroutine KORG2 Flow Chart

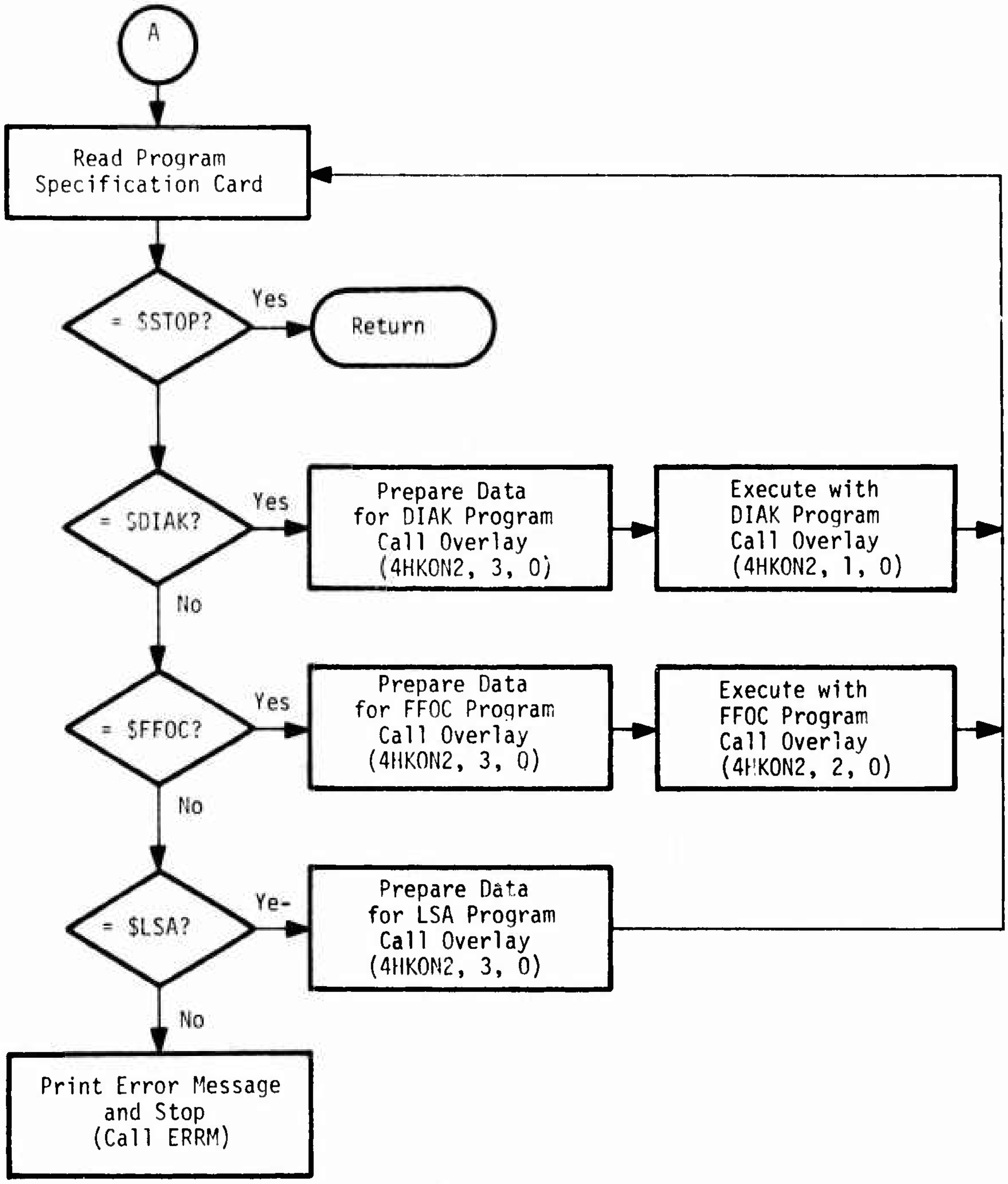

Figure 87. Subroutine KORG2 Flow Chart (Concluded)

```
      SUBROUTINE KORG2
C
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     PURPOSE - TO ORGANIZE EXECUTION OF KONPACT-2 PROGRAMS
C     DATE WRITTEN - JULY 1975
C
C     SUBPROGRAMS CALLED
C        IDRO
C        IDRP
C        ERRM
C
C     LABELLED COMMON LIST
C        IR                      FILE NO FOR INPUT DATA BUFFER
C        IW                      FILE NO FOR LINE PRINTER
C        IPRINT                  PRINT CONTROL FLAG
C        INSERT                  HOLLERITH INSE
C        LOCATE                  HOLLERITH LOCA
C        NULL                    HOLLERITH NULL
C        MARK                    HOLLERITH $$..$
C        JQ                      FILE NO FOR QUADRUPLE DATA FILE
C        JS                      FILE NO FOR SCRATCH FILE
C        JSD                     FILE NO FOR SDSTP FILE
C        JF                      FILE NO FOR FDATA FILE
C        JD                      FILE NO FOR DDATA FILE
C
      DIMENSION CARD(20)
      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),
     1JQ,JS,JSD,JF,JD
      COMMON /INF/ NXM,NPM,NUM,CODE
      INTEGER HINSE,HLOCA,HNULL,HDOLR
      DATA HINSE,HLOCA,HNULL,HDOLR/4HINSE,4HLOCA,4HNULL,4H$$$$/
      DATA HPRIN,HTHIN,HERYT/4HPRIN,4HTHIN,4HERYT/
      DATA HTPUT,HNAL,HPUT/4HTPUT,4HNAL ,4HPUT /
      DATA HODIA,HDFFO,HOLSA/4H$DIA,4H$FFO,4H$LSA/
      DATA HDSTG/4H$STG/
      DATA HC/1HC/
C
C     DEFINE INPUT/OUTPUT PARAMETERS
C
      IR=7 $ IW=9 $ IPRINT=4 $ JQ=8 $ JS=5 $ JSD=2 $ JF=1 $ JD=6
      INSERT=HINSE $ LOCATE=HLOCA $ NULL=HNULL
      DO 100 I=1,20
      MARK(I)=HDOLR
  100 CONTINUE
C
C     READ INPUT DATA INTO INPUT DATA BUFFER FILE
C
  110 CONTINUE
      READ(4,120)CARD
      IF(EOF(4))140,115
  115 CONTINUE
      WRITE(IR,120)CARD
  120 FORMAT(20A4)
  130 FORMAT(A1,A3)
      GO TO 110
  140 CONTINUE
      ENDFILE IR
      REWIND IR
C
C     READ PRINT SPECIFICATION AND SET IPRINT
C
  142 CONTINUE
      READ(IR,120) CARD
```

Figure 88. Subroutine KORG2 Program Listing

```
      DECODE(4,130,CARD(1))CC,DUMMY
      IF(CC.EQ.HC)GO TO 142
      IF(CARD(1).NE.HPRIN)GO TO 152
      IF(CARD(3).EQ.HTHIN)IPRINT=1
      IF(CARD(3).EQ.HTHIN)GO TO 142
      IF(CARD(3).EQ.HERYT)IPRINT=6
      IF(CARD(3).EQ.HERYT)GO TO 142
      IF(CARD(3).NE.HTPUT)GO TO 144
      IF(IPRINT.EQ.1)IPRINT=5
      IF(IPRINT.EQ.5)GO TO 142
      IPRINT=3
      GO TO 142
  144 CONTINUE
      IF(CARD(3).NE.HNAL)GO TO 146
      IF(IPRINT.EQ.1)IPRINT=4
      IF(IPRINT.EQ.4)GO TO 142
      IPRINT=2
      GO TO 142
  146 CONTINUE
      IF(CARD(3).NE.HPUT)GO TO 148
      IF(IPRINT.EQ.2)IPRINT=4
      IF(IPRINT.EQ.4)GO TO 142
      IF(IPRINT.EQ.3)IPRINT=5
      IF(IPRINT.EQ.5) GO TO 142
      IPRINT=1
      GO TO 142
  148 CONTINUE
C
C     PRINT ERROR MESSEGE
C
      WRITE(IW,150)
  150 FORMAT(1H1,//,1X,3 HPRINT CARD SPECIFICATION ERROR,//,1X,
     143HINPUT AND FINAL OUTPUT DATA WILL BE PRINTED)
  152 CONTINUE
      REWIND IR
C
C     PRINT INPUT DATA
C
      IF((IPRINT.NE.1).AND.(IPRINT.LT.4))GO TO 158
      WRITE(IW,154)
  154 FORMAT(1H1,//,1X,24H*** INPUT DATA CARDS ***,//)
      CALL IDPR(IR,IW)
      WRITE(IW,156)
  156 FORMAT(//,1X,31H*** END OF INPUT DATA CARDS ***,//)
      REWIND IR
  158 CONTINUE
C
C     REORGANIZE INPUT DATA
C
      CALL IDRO(IR,IW,JS)
C
C     PRINT REORGANIZED INPUT DATA
C
      IF(IPRINT.LT.6)GO TO 164
      WRITE(IW,160)
  160 FORMAT(1H1,//,1X,33H*** REORGANIZED INPUT DATA ***,//)
      CALL IDPR(IR,IW)
      WRITE(IW,162)
  162 FORMAT(//,1X,37H*** END OF REORGANIZED INPUT DATA ***,//)
  164 CONTINUE
C
C     READ INPUT DATA CARDS
C
  159 CONTINUE
      READ(IR,120) CARD
      IF(CARD(1).EQ.HPRIN) GO TO 159
```

Figure 88. Subroutine KORG2 Program Listing (Continued)

```
      IF(CARD(1).EQ.HDSTO) RETURN
      IF(CARD(1).EQ.HDDIA) GO TO 180
      IF(CARD(1).EQ.HDEFO)  GO TO 260
      IF(CARD(1).EQ.HDLSA) GO TO 360
      CALL ERRM(1,4HKORG,4H2   ,0.,1,IW)
C
C     CALL OVERLAY LOADER TO LOAD REQUIRED PROGRAMS FOR EXECUTION
C
  180 CONTINUE
      CODE=CARD(1)
      CALL OVERLAY(4HKON2,3,0)
      CALL OVERLAY(4HKON2,1,0)
      GO TO 159
  260 CONTINUE
      CODE=CARD(1)
      CALL OVERLAY(4HKON2,3,0)
      CALL OVERLAY(4HKON2,2,0)
      GO TO 159
  360 CONTINUE
      CODE=CARD(1)
      CALL OVERLAY(4HKON2,3,0)
      GO TO 159
      END
```

Figure 88. Subroutine KORG2 Program Listing (Concluded)

```
      SUBROUTINE TIMER(A,G2,HDK,X,XI,DX,DXI,XL     ,GN,GS,R,IPLR,ITITL,
     1IUNIT,     CL,T,DT,ST,YMAX,YMIN,IFLT,IRUN,IDATE,NSTEP,NRAND,NPLOT,
     2NPRIN,NN,NX,NF,NG,NC,NR,MXR,MN,MX,MXRSP,NOP,NAME1,NAME2,SCAL,NEWY,
     3T1,T2,NGLG)
      DIMENSION A(MX,MX),G2(MX,MN),HDK(MX,MX),X(MX),DX(MX),DXI(MX)
      DIMENSION XI(MX,MN),XL(MX,2)      ,GN(MX,MN),GS(MX,MN),R(MARSP)
      DIMENSION IPLR(MXR),ITITL(MXR),IUNIT(MXR),IBUF( 15),LW(10)
      DIMENSION YMAX(MXR),YMIN(MXR),CL(MN,1)
      DIMENSION SCAL (MXR)
      DIMENSION NEWY(48)
      INTEGER BLANK
      RUN=IRUN
      IF(NOP.GT.0) GO TO 22
      WRITE(9,203)
  203 FORMAT(1H1/7X,44HNO. OF PLOTS IS ZERO - IGNORE TIME RESPONSES)
      RETUR
   22 WRITE(9,102)
  102 FORMAT(1H1/7X,14HTIME RESPONSES//)
      BLANK=10H
      LW(1)=10H*
      LW(2)=10H *
      LW(3)=10H  *
      LW(4)=10H   *
      LW(5)=10H    *
      LW(6)=10H     *
      LW(7)=10H      *
      LW(8)=10H       *
      LW(9)=10H        *
      LW(10)=10H         *
C   NSTEP=0  NO STEP INPUTS
C        =1  STEP COMMANDS
C        =2  STEP GUSTS
C        =3  BOTH
C        =4  NO STEP INPUTS - TRANSIENTS ONLY
C   NRAND=0  NO RANDOM INPUTS
C   NRAND=1  GUSTS
C   NPRIN=0   DON'T PRINT RESPONSES
C   NPRIN=1   PRINT RESPONSES
C   NPLOT=0  NO PLOTS
C   NPLOT=1  CALCOMP PLOTS
C   NPLOT=2  LINE PRINTER PLOTS
C   NPLOT=3  BOTH
      IBUF(1)=IDATE
      IBUF(2)=BLANK
      IBUF(3)=NAME1
      IBUF(4)=NAME2
      IBUF(5)=10H    FLIGHT
      IBUF(6)=10H CONDITION
      IF(NPLOT.EQ.0.OR.NPLOT.EQ.2) GO TO 2
      CALL FACTOR(2.0)
      CALL PLOT(0.,-13.,-3)
      CALL PLOT(0.,0.5,-3)
      CALL SYMBOL(0.,0.,.14,IBUF(1),90.,40)
      CALL SYMBOL(.5,0.,.14,IBUF(5),90.,20)
      CALL SYMBOL(.5,3.,.14,IFLT,90.,10)
      CALL SYMBOL(1.,0.,.14,3HRUN,90.,3)
      CALL NUMBER(1., .6,.14, RUN,90.,-1)
      CALL PLOT(1.5,0.,-3)
      IF(NPLOT.EQ.1) GO TO 1
    2 WRITE(9,100) (IBUF(I),I=1,6),IFLT,IRUN
100   FORMAT(2X,6A10,2X,A10,2X,3HRUN,I3//)
    1 CONTINUE
      DO 3 I=1,NX
```

Figure 89. Subroutine TIMER Program Listing

```
      DO 3 J=1,NN
      GN(I,J)=0.
    3 GS(I,J)=0.
      NEG=NX
      IF(NSTEP.EQ.4) GO TO 7
      IF(NRAND.EQ.0) GO TO 4
      N=NF+1
      NEG=NX-NC
      ETA=GRAN(1)
      DO 5 I=N,NEG
      DO 5 J=1,NN
    5 GN(I,J)=G2(I,J)/SQRT(DT)
      GO TO 10
    4 NEG=NF
   10 IF(NSTEP.EQ.2) GO TO 6
      IF(NSTEP.EQ.0) GO TO 7
      DO 8 I=1,NF
      DO 8 J=1,NC
      JG=J+NG
      JJ=J+NX-NC
    8 GS(I,JG)=A(I,JJ)*CL(JG,1)
      IF(NSTEP.NE.3) GO TO 7
6     NEG=NF+NGLG
    7 CONTINUE
      WRITE(9,101) NOP
  101 FORMAT(7X,9HTHERE ARE,I3,21H RESPONSES TO COMPUTE//)
      NT=T/DT
      S=NT
      S=S*DT
      IF(S.LT.T) NT=NT+1
      NT=NT+1
      NTP=ST/DT
      S=NTP
      S=S*DT
      IF(S.LT.ST) NTP=NTP+1
      IF(NTP.EQ.0) NTP=1
      NTTP=NT/NTP
      NTTP=NTTP+1
      NPTOT=0
      DO 12 J=1,NN
      IF(NSTEP.EQ.4) GO TO 51
      IF(J.GT.NG) GO TO 41
      IF(NRAND.EQ.0.AND.NSTEP.EQ.1) GO TO 12
      GO TO 51
41    IF(NSTEP.EQ.0.OR.NSTEP.EQ.2) GO TO 12
   51 DO 11 I=1,NX
      X(I)=XI(I,J)
   11 DX1(I)=0.
      IF(J.LE.NG) GO TO 56
      IF(NSTEP.EQ.0) GO TO 56
      IF(NSTEP.EQ.2) GO TO 56
      IF(NSTEP.EQ.4) GO TO 56
      JJ=J+NX-NC-NG
      X(JJ)=CL(J,1)
56    CONTINUE
      IF(NPRIN .EQ.0) GO TO 24
      WRITE(9,103) J
  103 FORMAT(1H1/7X,30HTIME RESPONSES FOR DISTURBANCE,I3//)
   24 DO 17 IT=1,NT
      IF(J.GT.NG) GO TO 13
      IF(NSTEP.LE.1) GO TO 55
      IF(NSTEP.GT.3) GO TO 55
      CALL SGUST(A,GS,CL,X,DT,T1,T2,J,NEG,NG,IT,NX,MU)
55    CONTINUE
      IF(NRAND.EQ.0) GO TO 13
      ETA=GRAN(1)
```

Figure 89. Subroutine TIMER Program Listing (Continued)

```
      GO TO 14
   13 ETA=0.
   14 DO 15 I=1,NEG
      DX(I)=GN(I,J)*ETA+GS(I,J)
      DO 52 K=1,NEG
   52 DX(I)=DX(I)+A(I,K)*X(K)
      IF(XL(I,2).LE.0.) GO TO 15
      IF(ABS(DX(I)).GT.XL(I,2))DX(I)=SIGN(XL(I,2),DX(I))
   15 CONTINUE
      DO 16 I=1,NEG
      X(I)=X(I)+DT*(3.*DX(I)-DX1(I))/2.
      DX1(I)=DX(I)
      IF(XL(I,1).LE.0.) GO TO 16
      IF(ABS(X(I)).GT.XL(I,1)) X(I)=SIGN(XL(I,1),X(I))
   16 CONTINUE
      IF(NTP.EQ.1) GO TO 18
      ITM=MOD(IT,NTP)
      IF(ITM.NE.1) GO TO 17
      ITT=(IT/NTP)+1
   18 IF(NTP.EQ.1) ITT=IT
      II=ITT-NTTP-2
      IP=1
      DO 19 I=1,NR
      IF(IP.GT.NOP) GO TO 23
      IF(IPLR(IP).NE.I) GO TO 19
      II=II+NTTP+2
      R (II)=0.
      DO 20 K=1,NX
   20 R(II)=R(II)+HDK(I,K)*X(K)
      R(II)=R(II)*SCAL  (IP)
      IP=IP+1
   19 CONTINUE
      DO 21 I=1,NEG
      IX=I+NR
      IF(IP.GT.NOP) GO TO 23
      IF(IPLR(IP).NE.IX) GO TO 21
      II=II+NTTP+2
      R(II)=X(I)*SCAL  (IP)
      IP=IP+1
   21 CONTINUE
   23 II=II+NTTP+2
      TIME=ITT-1
      TIME=TIME*ST
      R(II)=TIME
      IF(NPRIN .EQ.0) GO TO 17
      WRITE(9,202) TIME
  202 FORMAT(5X,6HTIME =,F10.3)
  200 FORMAT(5X,4(A10,2H= ,G8.2,2X))
      IPR=ITT
      K =-3
      DO 26 INP=1,NOP ,4
      K =K +4
      IPR1=IPR+NTTP+2
      IPR2=IPR1+NTTP+2
      IPR3=IPR2+NTTP+2
      KK=K+3-NOP
      IF(KK.LE.0) GO TO 54
      IF(KK.EQ.3) WRITE(9,200) ITITL(K),R(IPR)
      IF(KK.EQ.2) WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1)
      IF(KK.EQ.1) WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1)
     1,ITITL(K+2),R(IPR2)
      GO TO 26
54    WRITE(9,200) ITITL(K), R(IPR),ITITL(K+1),R(IPR1),ITITL(K+2),
     1R(IPR2),ITITL(K+3),R(IPR3)
   26 IPR=IPR3+NTTP+2
      WRITE(9,201)
```

Figure 89. Subroutine TIMER Program Listing (Continued)

```
201 FORMAT(1H0)
 17 CONTINUE
    IF(NPLOT.EQ.0) GO TO 12
    IF(NPLOT.EQ.1) GO TO 30
    WRITE(9,103) J
    DO 32 K=1,NOP
    DO 31 I=1,15
 31 IBUF(I)=BLANK
    IF(NEWY(K).EQ.0) GO TO 33
    YMAX(K)=-1.E+20
    YMIN(K)= 1.E+20
    DO 34 L=1,NTTP
    I=(K-1)*(NTTP+2)+L
    YMAX(K)=AMAX1(YMAX(K),R(I))
 34 YMIN(K)=AMIN1(YMIN(K),R(I))
 33 CONTINUE
    IF(YMIN(K).EQ.YMAX(K)) GO TO 32
    RANGE=YMAX(K)-YMIN(K)
    IBUF(3)=ITITL(K)
    IBUF(5)=IUNIT(K)
    WRITE(9,104) (IBUF(I),I=1,5)
104 FORMAT(1H1/5A10//)
    IBUF(3)=BLANK
    IBUF(5)=BLANK
    X(1)=YMIN(K)
    DO 36 I=1,5
 36 X(I+1)=X(I)+RANGE/5.
    WRITE(9,105) (X(I),I=1,6)
105 FORMAT(6F14.3)
    WRITE(9,106)
106 FORMAT(7X,2H I,5(14H-------------I))
    DO 37 L=1,NTTP
    I=(K-1)*(NTTP+2)+L
    IF(RANGE.EQ.0.) GO TO 42
    LL=(R(I)-YMIN(K))*70./RANGE
    IF(LL.EQ.0) GO TO 42
    INL=LL/10
    IF(INL.LE.0) INL=0
    IF(INL.GT.11)INL=11
    IW=MOD(LL,10)
    IF(IW.LE.0) GO TO 40
    IF(INL.EQ.0) GO TO 53
    DO 38 I=1,INL
 38 IBUF(I)=BLANK
53  IBUF(INL+1)=LW(IW)
    IB=INL+2
    DO 39 I=IB,15
 39 IBUF(I)=BLANK
    GO TO 45
40  IW=10
    IW=10
    IE=INL-1
    DO 43 I=1,IE
 43 IBUF(I)=BLANK
    IBUF(IE+1)=LW(IW)
    IB=IE+2
    DO 44 I=IB,15
 44 IBUF(I)=BLANK
    GO TO 45
 42 IBUF(1)=LW(1)
    DO 46 I=2,15
 46 IBUF(I)=BLANK
 45 TIME =L-1
    TIME=TIME*ST
    WRITE(9,107) TIME,(IBUF(I),I=1,12)
107 FORMAT(F6.2,3HS I,12A10)
```

Figure 89. Subroutine TIMER Program Listing (Continued)

```
      DO 47 I=1,15
   47 IBUF(I)=BLANK
   37 CONTINUE
   32 CONTINUE
   30 CONTINUE
      IF(NPLOT.EQ.2) GO TO 12
      FJ=J
      CALL SYMBOL(0.,0.,.14,30HTIME RESPONSES FOR DISTURBANCE,90.,30)
      CALL NUMBER(0.,5.,.14,FJ,90.,-1)
      CALL PLOT(1.,0.,-3)
      IARX=NOP *(NTTP+2)+1
      DO 48 K=1,NOP
      IF(NEWY(K).EQ.0) GO TO 49
      IAR=(K-1)*(NTTP+2)+1
      CALL SCALE(R(IAR),8.,NTTP,1)
      CALL SCALE(R(IARX),10.,NTTP,1)
      GO TO 50
   49 IAR=K*(NTTP+2)-1
      IF(YMIN(K).EQ.YMAX(K)) GO TO 48
      R(IAR)=YMIN(K)
      R(IAR+1)=(YMAX(K)-YMIN(K))/8.
      IAR=IARX+NTTP
      R(IAR)=0.
      R(IAR+1)=T/10.
   50 IBUF(1)=ITITL(K)
      IBUF(2)=BLANK
      IBUF(3)=IUNIT(K)
      IBUF(4)=10HTIME IN SE
      IBUF(5)=10HCONDS
      IAR=K*(NTTP+2)-1
      IARP=IAR+1
      CALL AXIS(0.,0.,IBUF(1),30,8.,90.,R(IAR),R(IARP))
      IAR=IARX+NTTP
      IARP=IAR+1
      CALL AXIS(0.,0.,IBUF(4),-20,10.,0.,R(IAR),R(IARP))
      IAR=(K-1)*(NTTP+2)+1
      CALL LINE(R(IARX),R(IAR),NTTP,1,0,0)
      CALL PLOT(13.,0.,-3)
      NPTOT=NPTOT+1
      IF(NPTOT.LT.5) GO TO 48
      NPTOT=0
      CALL DSP(2)
      CALL PLOT(0.,-13.,-3)
      CALL PLOT(0.,0.5,-3)
      IBUF(1)=IDATE
      IBUF(2)=BLANK
      IBUF(3)=NAME1
      IBUF(4)=NAME2
      IBUF(5)=10H    FLIGHT
      IBUF(6)=10H CONDITION
      CALL SYMBOL(0.,0.,.14,IBUF(1),90.,40)
      CALL SYMBOL(.5,0.,.14,IBUF(5),90.,20)
      CALL SYMBOL(.5,3.,.14,IFLT,90.,10)
      CALL SYMBOL(1.,0.,.14,3HRUN,90.,3)
      CALL NUMBER(1., .6,.14, RUN,90.,-1)
      CALL PLOT(1.5,0.,-3)
   48 CONTINUE
   12 CONTINUE
      RETURN
      END
```

Figure 89. Subroutine TIMER Program Listing (Concluded)

```
      SUBROUTINE SGUST(A,GS,CL,X,DT,T1,T2,J,NF,NG,IT,MX,MN)
      DIMENSION A(MX,MX),GS(MX,MN),CL(MN,1),X(MX)
      IF(IT.GT.1) GO TO 3
      JJ=J+NF
      DO 5 I=1,NF
    5 GS(I,J)=A(I,JJ)*CL(J,1)
      X(JJ)=CL(J,1)
      ND1=T1/DT
      ND2=T2/DT
      S=ND1*DT
      IF(S.LT.T1) ND1=ND1+1
      S=ND2*DT
      IF(S.LT.T2) ND2=ND2+1
      ND1=ND1+1
      ND2=ND2+1
    3 IF(IT.LT.ND1) RETURN
      IF(IT.GT.ND1) GO TO 2
      JJ=J+NG+NF
      DO 1 I=1,NF
    1 GS(I,J)=GS(I,J)+A(I,JJ)*CL(J,1)
      X(JJ)=CL(J,1)
    2 IF(IT.NE.ND2) RETURN
      JJ=J+2*NG+NF
      DO 4 I=1,NF
    4 GS(I,J)=GS(I,J)+A(I,JJ)*CL(J,1)
      X(JJ)=CL(J,1)
      RETURN
      END
```

Figure 90. Subroutine SGUST Program Listing

```
      SUBROUTINE CAL1(A,XN,P,KWA,N,NR,IMAX,IT,IERR,EE)
      DIMENSION A(NR,1),XN(NR,1),P(NR,1),KWA(NR)
      IERR=
      TR=0.
      DO 300 I=1,N
  300 TR=TR+A(I,I)
      FN=N
      TR=AMAX1(TR,-FN)
      IF(TR)301,2,2
2     IERR=1
      GO TO 601
  301 ALF=ABS(TR)/FN
      NC=N*(N+1)
      NC=NC/2
      DO 60 I=1,N
      DO 63 J=1,N
      GOTO(61,62),IT
   61 P(I,J)=A(I,J)
      GOTO 63
   62 P(I,J)=A(J,I)
   63 CONTINUE
      P(I,I)=P(I,I)-ALF
   60 CONTINUE
      CALL TDINVR(ISOL,IDSOL,N,N,P,NR,KWA,DET)
      IF((ISOL+IDSOL).LE.2) GO TO 22
      IERR=3
      GO TO 601
   22 DO 4 I=1,N
      DO 4 J=1,N
      A(I,J)=0.
      DO 4 K=1,N
    4 A(I,J)=A(I,J)+P(K,I)*XN(K,J)*2.*ALF
      DO 5 I=1,N
      DO 5 J=1,N
      XN(I,J)=0.
      DO 5 K=1,N
    5 XN(I,J)=XN(I,J)+A(I,K)*P(K,J)
      DO 7 I=1,N
      DO 8 J=1,N
    8 P(I,J)=P(I,J)*2.*ALF
    7 P(I,I)=P(I,I)+1.
      ITER=
  100 CONTINUE
      DO 9 I=1,N
      DO 9 J=1,N
      A(I,J)=0.
      DO 9 K=1,N
    9 A(I,J)=A(I,J)+P(K,I)*XN(K,J)
      ICOT=
      DO 10 I=1,N
      DO 10 J=I,N
      DXIJ= .
      DO 11 K=1,N
   11 DXIJ=DXIJ+A(I,K)*P(K,J)
      XN(I,J)=XN(I,J)+DXIJ
      XN(J,I)=XN(I,J)
      AXN=ABS(XN(I,J))
      IF(AXN.LT.1.E-20) GO TO 14
      IF(AXN.LT.1.E 20) GO TO 201
      IERR=2
      GO TO 601
  201 RAT=ABS(DXIJ/XN(I,J))
      IF(RAT-EE)14,14,70
```

Figure 91. Subroutine CAL1 Program Listing

```
 14 ICOT=ICOT+1
 70 CONTINUE
 10 CONTINUE
 18 ITER=ITER+1
    IF(ICOT-NC)15,50,15
 15 CONTINUE
    DO 20 I=1,N
    DO 20 J=1,N
 20 A(I,J)=P(I,J)
 16 DO 17 I=1,N
    DO 17 J=1,N
    P(I,J)=0.
    DO 17 K=1,N
 17 P(I,J)=P(I,J)+A(I,K)*A(K,J)
 40 IF(ITER-IMAX)100,50,50
 50 CONTINUE
    WRITE(9,600) ITER
600 FORMAT(/7X,6H ITER=I2)
    RETURN
601 WRITE(9,602) IERR
602 FORMAT(/7X,6H IERR=I2)
    RETURN
    END
```

Figure 91. Subroutine CAL1 Program Listing (Concluded)

```
      SUBROUTINE STRIC(A,B,PS,W,S,TPF,NX,NU,NXM,MU)
      DIMENSION A(NXM,NXM),B(NXM,MU),W(NXM,NXM),S(NXM,NXM)
      DIMENSION TPF(NXM,NXM),PS(NXM,NXM)
      DT=.01
      DO 1 I=1,NX
      DO 1 J=1,NX
      W(I,J)=0.
      DO 1 K=1,NU
    1 W(I,J)=W(I,J)+B(I,K)*B(J,K)*DT
      NT=10
      KT=100
      T=0.
      DO 200 L=1,KT
      T=T+DT
      DO 3 I=1,NX
      DO 4 J=1,NX
      S(I,J)=0.
    4 TPF(I,J)=0.
      S(I,I)=1.
    3 TPF(I,I)=1.
      DO 100 M=2,NT
      FAC=M-1
      FAC=1./FAC
      DO 5 I=1,NX
      DO 5 J=1,NX
      PS(I,J)=0.
      DO 6 K=1,NX
    6 PS(I,J)=PS(I,J)-TPF(I,K)*A(K,J)*FAC*T
    5 S(I,J)=S(I,J)+PS(I,J)
      DO 7 I=1,NX
      DO 7 J=1,NX
    7 TPF(I,J)=PS(I,J)
  100 CONTINUE
      DO 8 I=1,NX
      DO 8 J=1,NU
      TPF(I,J)=0.
      DO 8 K=1,NX
    8 TPF(I,J)=TPF(I,J)+S(I,K)*B(K,J)
      DO 9 I=1,NX
      DO 9 J=1,NX
      DO 9 K=1,NU
    9 W(I,J)=W(I,J)+TPF(I,K)*TPF(J,K)*DT
  200 CONTINUE
      WRITE(9,300)
  300 FORMAT(1H1/7X,12H W(T) MATRIX/)
      CALL MP(NXM,NXM,NX,NX,W)
      RETURN
      END
```

Figure 92. Subroutine STRIC Program Listing

```
      SUBROUTINE SHUFL(A,MM,NN,M,N,MC,NC,NORD,B,MX)
      DIMENSION A(MM,NN),NORD(MX),B(MX,MX)
      IF(MC.EQ.0) GO TO 1
      DO 2 I=1,M
      II=NORD(I)
      DO 2 J=1,N
    2 B(I,J)=A(II,J)
      DO 3 I=1,M
      DO 3 J=1,N
    3 A(I,J)=B(I,J)
    1 CONTINUE
      IF(NC.EQ.0) RETURN
      DO 4 J=1,N
      JJ=NORD(J)
      DO 4 I=1,M
    4 B(I,J)=A(I,JJ)
      DO 5 I=1,M
      DO 5 J=1,N
    5 A(I,J)=B(I,J)
      RETURN
      END
```

Figure 93. Subroutine SHUFL Program Listing

```
      SUBROUTINE SHUF(F,G1,G2,H,AM,AKG,Y,NORD,MX,NX,MR,NR,MM,NM,MU,NU,
     1MNN,NN)
      DIMENSION F(MX,MX),G1(MX,MU),G2(MX,MNN),H(MR,MX),AM(MM,MX),
     1AKG(MU,MX),NORD(MX)
      DIMENSION Y(MX,MX)
      DO 1 I=1,NX
      II=NORD(I)
      DO 1 J=1,NX
      JJ=NORD(J)
    1 Y(I,J)=F(II,JJ)
      DO 2 I=1,NX
      DO 2 J=1,NX
    2 F(I,J)=Y(I,J)
      DO 3 I=1,NX
      II=NORD(I)
      DO 4 J=1,NU
    4 Y(I,J)=G1(II,J)
      DO 3 J=1,NN
      JJ=J+NU
    3 Y(I,JJ)=G2(II,J)
      DO 5 I=1,NX
      DO 6 J=1,NU
    6 G1(I,J)=Y(I,J)
      DO 5 J=1,NN
      JJ=J+NU
    5 G2(I,J)=Y(I,JJ)
      DO 7 J=1,NX
      JJ=NORD(J)
      DO 7 I=1,NR
    7 Y(I,J)=H(I,JJ)
      DO 8 J=1,NX
      DO 8 I=1,NR
    8 H(I,J)=Y(I,J)
      DO 9 J=1,NX
      JJ=NORD(J)
      DO 9 I=1,NM
    9 Y(I,J)=AM(I,JJ)
      DO 10 J=1,NX
      DO 10 I=1,NM
   10 AM(I,J)=Y(I,J)
      DO 11 J=1,NX
      JJ=NORD(J)
      DO 11 I=1,NU
   11 Y(I,J)=AKG(I,JJ)
      DO 12 J=1,NX
      DO 12 I=1,NU
   12 AKG(I,J)=Y(I,J)
      RETURN
      END
```

Figure 94. Subroutine SHUF Program Listing

```
      SUBROUTINE RESP(C,A,G2,AM,AK,X,Y,Z,S,R,ES,E,U,V,XI,DQ,AKG,DQD,HDK,
     1KWA,NX,NFF,NN,NM,NU,NR,MX,MFF,MFB,MN,MM,MU,MR,ITER,IMAX,IERR,NCOV)
      DIMENSION XI(MFF,MFF), X(MX,MX),R(MX,MX),HDK(MR,MX),C(MX,MX),
     1G2(MX,MN),A(MX,MX),AM(MM,MX),AK(MU,MM),Y(MX,MX),Z(MX,MX),S(MX,MX),
     2ES(MX,MX),U(MFB,MFB),V(MFF,MFF),E(MFF,MFF),DQ(MU,MM),
     3KWA(MX),AKG(MU,MX),DQD(MU,MU)
      NFB=NX-NFF
C
C   COMPUTE COVARIANCE MATRIX FOR DISTURBANCE KCOM
      DO 6080 I=1,NR
      DO 6080 J=1,NR
 6080 R(I,J)=0.
      KCOM=0
 6076 KCOM=KCOM+1
      DO 4020 I=1,NFF
      DO 4020 J=1,NFF
      II=I+NFB
      JJ=J+NFB
 4020 C(I,J)=G2(II    ,KCOM)*G2(JJ    ,KCOM)
      WRITE(9,41) KCOM
   41 FORMAT(1H1/7X,36H COVARIANCE ANALYSIS FOR DISTURBANCE,I3//)
      ITER=0
      CALL COVAR(XI,A,C,X,G2,S,E,ES,V,U,NX,NFF,NN,MX,MFF,MFB,MN,IMAX,
     1ITER,2,IERR,KWA)
      IF(IERR.EQ.0) GO TO 896
      WRITE(9,43)
   43 FORMAT(1H1/7X,28H COVARIANCE MATRIX UNDEFINED//7X,27H IGNORE COVAR
     1IANCE ANALYSIS//)
      RETURN
  896 WRITE(9,4051)
 4051 FORMAT(//7X,18H COVARIANCE MATRIX//)
      CALL MP(MX,MX,NX,NX,X)
C
C   COMPUTE (H+DKM)X(H+DKM)
      DO 4053 I=1,NR
      DO 4053 J=1,NX
      C(I,J)=0.
      DO 4053 K=1,NX
 4053 C(I,J)=C(I,J)+HDK(I,K)*X(K,J)
      DO 4054 I=1,NR
      DO 4054 J=1,NR
      S(I,J)=0.
      DO 4054 K=1,NX
 4054 S(I,J)=S(I,J)+C(I,K)*HDK(J,K)
      IF(NCOV.GT.2) GO TO 2
      WRITE(9,42)
 42   FORMAT(1H1/7X,27H RESPONSE COVARIANCE MATRIX//)
      CALL MP(MX,MX,NR,NR,S)
      DO 7015 I=1,NX
      DO 7015 J=1,NM
      ES(J,I)=0.
      DO 7015 K=1,NX
 7015 ES(J,I)=ES(J,I)+X(I,K)*AM(J,K)
      DO 7016 I=1,NM
      DO 7016 J=1,NM
      Y(I,J)=0.
      DO 7016 K=1,NX
 7016 Y(I,J)=Y(I,J)+AM(I,K)*ES(J,K)
      WRITE(9,44)
   44 FORMAT(1H1/7X,30H MEASUREMENT COVARIANCE MATRIX//)
      CALL MP(MX,MX,NM,NM,Y)
      DO 1112 I=1,NU
      DO 1112 J=1,NM
```

Figure 95. Subroutine RESP Program Listing

```
      DQ(I,J)=0.
      DO 1112 K=1,NM
 1112 DQ(I,J)=DQ(I,J)+AK(I,K)*Y(K,J)
      DO 6085 I=1,NU
      DO 6085 J=1,NU
      DQD(I,J)=0.
      DO 6085 K=1,NM
 6085 DQD(I,J)=DQD(I,J)+DQ(I,K)*AK(J,K)
      WRITE(9,45)
   45 FORMAT(1H1/7X,26H CONTROL COVARIANCE MATRIX//)
      CALL MP(MU,MU,NU,NU,DQD)
      DO 1111 I=1,NX
      DO 1111 J=1,NX
      Z(I,J)=0.
      IF(X(I,I).LT.1.E-20) GO TO 1111
      IF(X(J,J).LT.1.E-20) GO TO 1111
      Z(I,J)=X(I,J)/SQRT(X(I,I)*X(J,J))
 1111 CONTINUE
      WRITE(9,46)
   46 FORMAT(1H1/7X,31H STATE CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NX,NX,Z)
      DO 1122 I=1,NU
      DO 1122 J=1,NX
      AKG(I,J)=0.
      DO 1122 K=1,NM
 1122 AKG(I,J)=AKG(I,J)+AK(I,K)*AM(K,J)
      DO 1113 I=1,NU
      DO 1113 J=1,NX
      Z(I,J)=0.
      IF(DQD(I,I).LT.1.E-20) GO TO 1113
      IF(X(J,J).LT.1.E-20) GO TO 1113
      DO 1123 K=1,NX
 1123 Z(I,J)=Z(I,J)+AKG(I,K)*X(K,J)
      Z(I,J)=Z(I,J)/SQRT(DQD(I,I)*X(J,J))
 1113 CONTINUE
      WRITE(9,47)
   47 FORMAT(1H1/7X,39H CONTROL-STATE CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NU,NX,Z)
      DO 1114 I=1,NR
      DO 1114 J=1,NX
      Z(I,J)=0.
      IF(S(I,I).LT.1.E-20) GO TO 1114
      IF(X(J,J).LT.1.E-20) GO TO 1114
      Z(I,J)=C(I,J)/SQRT(S(I,I)*X(J,J))
 1114 CONTINUE
      WRITE(9,48)
   48 FORMAT(1H1/7X,40H RESPONSE-STATE CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NR,NX,Z)
      DO 1115 I=1,NM
      DO 1115 J=1,NM
      Z(I,J)=0.
      IF(Y(I,I).LT.1.E-20) GO TO 1115
      IF(Y(J,J).LT.1.E-20) GO TO 1115
      Z(I,J)=Y(I,J)/SQRT(Y(I,I)*Y(J,J))
 1115 CONTINUE
      WRITE(9,49)
   49 FORMAT(1H1/7X,37H MEASUREMENT CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NM,NM,Z)
      DO 1300 I=1,NM
      DO 1300 J=1,NX
      Z(I,J)=0.
      IF(Y(I,I).LT.1.E-20) GO TO 1300
      IF(X(J,J).LT.1.E-20) GO TO 1300
      DO 1301 K=1,NX
 1301 Z(I,J)=Z(I,J)+AM(I,K)*X(K,J)
      Z(I,J)=Z(I,J)/SQRT(Y(I,I)*X(J,J))
```

Figure 95. Subroutine RESP Program Listing (Continued)

```
1300  CONTINUE
      WRITE(9,1302)
1302  FORMAT(1H1/7X,43H MEASUREMENT-STATE CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NX,NX,Z)
      DO 1116 I=1,NU
      DO 1116 J=1,NM
      Z(I,J)=0.
      IF(DQD(I,I).LT.1.E-20) GO TO 1116
      IF(Y(J,J).LT.1.E-20) GO TO 1116
      Z(I,J)=DQ(I,J)/SQRT(DQD(I,I)*Y(J,J))
1116  CONTINUE
      WRITE(9,50)
   50 FORMAT(1H1/7X,45H CONTROL-MEASUREMENT CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NU,NM,Z)
      DO 1118 I=1,NR
      DO 1118 J=1,NM
      Z(I,J)=0.
      IF(S(I,I).LT.1.E-20) GO TO 1118
      IF(Y(J,J).LT.1.E-20) GO TO 1118
      DO 1119 K=1,NX
1119  Z(I,J)=Z(I,J)+C(I,K)*AM(J,K)
      Z(I,J)=Z(I,J)/SQRT(S(I,I)*Y(J,J))
1118  CONTINUE
      WRITE(9,51)
   51 FORMAT(1H1/7X,46H RESPONSE-MEASUREMENT CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NR,NM,Z)
      DO 1120 I=1,NU
      DO 1120 J=1,NR
      Z(I,J)=0.
      IF(DQD(I,I).LT.1.E-20) GO TO 1120
      IF(S(J,J).LT.1.E-20) GO TO 1120
      DO 1121 K=1,NX
1121  Z(I,J)=Z(I,J)+AKG(I,K)*C(J,K)
      Z(I,J)=Z(I,J)/SQRT(DQD(I,I)*S(J,J))
1120  CONTINUE
      WRITE(9,52)
   52 FORMAT(1H1/7X,42H CONTROL-RESPONSE CROSS-CORRELATION MATRIX//)
      CALL MP(MX,MX,NU,NR,Z)
2     DO 4056 I=1,NR
      IF(S(I,I).LT.0.) S(I,I)=0.
      R(I,I)=R(I,I)+S(I,I)
4056  S(I,I)=SQRT(S(I,I))
      WRITE(9,4057)((I,S(I,I)),I=1,NR)
4057  FORMAT(//20X,17H R.M.S. RESPONSES/(18X,I3,E16.8))
      IF(KCOM.LT.NN) GO TO 6076
      DO 6082 I=1,NR
6082  R(I,I)=SQRT(R(I,I))
      WRITE(9,6081)((I,R(I,I)),I=1,NR)
6081  FORMAT(//7X,22HTOTAL R.M.S. RESPONSES/(18X,I3,E16.8))
      RETURN
      END
```

Figure 95. Subroutine RESP Program Listing (Concluded)

```
      SUBROUTINE COVAR(XI,A,C,X,G2,S,E,ES,V,U,NX,NFF,NN,MX,MFF,MFB,MN,
     1IMAX,ITER,IR,IERR,KWA)
      DIMENSION A(MX,MX),C(MX,MX),G2(MX,MN),X(MX,MX),S(MX,MX),XI(MFF,MFF
     1),KWA(MX),E(MFF,MFF),ES(MX,MX),U(MFB,MFB),V(MFF,MFF)
      NFB=NX-NFF
      IF(ITER.NE.0) GO TO 150
C
C     COVARIANCE CALCULATION
C
C
C     COMPUTE  X22  FROM 0=A22*X22+X22*A22+G22*G22
    9 DO 11 I=1,NFF
      DO 11 J=1,NFF
      II=I+NFB
      JJ=J+NFB
      S(I,J)=A(II,JJ)
      IF(IR.EQ.2) GO TO 11
      C(I,J)=0.
      DO 12 K=1,NN
   12 C(I,J)=C(I,J)+G2(II    ,K)*G2(JJ    ,K)
11    CONTINUE
      CALL CAL(S,C,X,KWA,NFF,MX,IMAX,2,IERR)
      IF(IERR.GT.0) RETURN
      DO 151 I=1,NFF
      DO 151 J=1,NFF
  151 XI(I,J)=C(I,J)
C
C     COMPUTE  X12  FROM 0=A11*X12+X12*A22+A12*X22
150   DO 152 I=1,NFB
      DO 1 L=1,NFB
    1 X(I,L)=A(I,L)
      DO 152 J=1,NFF
      C(I,J)=0.
      DO 152 K=1,NFF
      KK=K+NFB
  152 C(I,J)=C(I,J)+A(I,KK)*XI(K,J)
      DO 153 I=1,NFF
      DO 153 J=1,NFF
      II=I+NFB
      JJ=J+NFB
  153 S(I,J)=A(JJ,II)
      CALL GCAL(S,X,C,NFF,NFB,MFF,MFB,IMAX,E,ES,V,U,KWA,MX,IERR)
      IF(IERR.GT.0) RETURN
C
C     COMPUTE  X11  FROM 0=A11*X11+X11*A11+A12*X12+X12*A12
      DO 154 I=1,NFB
      DO 154 J=1,NFB
      X(I,J)=A(I,J)
      S(I,J)=0.
      DO 154 K=1,NFF
      KK=K+NFB
  154 S(I,J)=S(I,J)+A(I,KK)*C(J,K)+C(I,K)*A(J,KK)
      CALL CAL(X,S,ES,KWA,NFB,MX,IMAX,2,IERR)
      IF(IERR.GT.0) RETURN
      DO 155 I=1,NFB
      DO 155 J=1,NFF
      JJ=J+NFB
      X(I,JJ)=C(I,J)
155   X(JJ    ,I)=X(I,JJ)
      DO 156 I=1,NFF
      DO 156 J=1,NFF
      II=I+NFB
      JJ=J+NFB
```

Figure 96. Subroutine COVAR Program Listing

```
156 X(II,IJ)=XI(I,J)
    DO 157 I=1,NFR
    DO 157 J=1,NFR
157 X(I,J)=S(I,J)
    RETURN
    END
```

Figure 96. Subroutine COVAR Program Listing (Concluded)

```
      SUBROUTINE COSTAT(R,A,S,X,ES,Y,Z,E,U,V,KWA,MX,MFB,MFF,NX,NFF,IMAX,
     1IERR)
      DIMENSION Y(MX,MX),A(MX,MX),S(MX,MX),Z(MX,MX),ES(MX,MX),X(MX,MX),
     1U(MFB,MFB),R(MX,MX),V(MFF,MFF),KWA(MX),E(MFF,MFF)
      NFB=NX-NFF
      DO 1 I=1,NX
      DO 1 J=1,NX
      S (I,J)=R(I,J)
    1 X(I,J)=A(I,J)
C
C      COMPUTE  S11   FROM 0=S11*A11+A11*S11+R11
      CALL CAL(X,S,ES,KWA,NFB,MX,IMAX,1,IERR)
      IF(IERR.GT.0) RETURN
C
C   COMPUTE S12 FROM 0=S12XA11+A22*S12+S11*A12+R12
      DO 158 I=1,NFB
      DO 158 J=1,NFF
      JJ=J+NFB
      ES(I,J)=R(I,JJ)
      DO 158 K=1,NFB
  158 ES(I,J)=ES(I,J)+S(I,K)*A(K,JJ)
      DO 159 I=1,NFB
      DO 159 J=1,NFB
  159 X(I,J)=A(J,I)
      DO 160 I=1,NFF
      DO 160 J=1,NFF
      II=I+NFB
      JJ=J+NFB
  160 Y(I,J)=A(II,JJ)
      CALL GCAL(Y,X,ES,NFF,NFB,MFF,MFB,IMAX,E,Z,V,U,KWA,MX,IERR)
      IF(IERR.GT.0) RETURN
      DO 162 I=1,NFB
      DO 162 J=1,NFF
      JJ=J+NFB
      S(JJ,I)=ES(I,J)
  162 S(I,JJ)=ES(I,J)
C
C     COMPUTE S22 FROM 0=S22*A22+A22*S22+A12*S12+S12*A12+R22
      DO 163 I=1,NFF
      DO 163 J=1,NFF
      II=I+NFB
      JJ=J+NFB
      Y(I,J)=A(II,JJ)
      X(I,J)=R(II,JJ)
      DO 163 K=1,NFB
  163 X(I,J)=X(I,J)+A(K,II)*S(K,JJ)+S(K,II)*A(K,JJ)
      CALL CAL(Y,X,R,KWA,NFF,MX,IMAX,1,IERR)
      IF(IERR.GT.0) RETURN
      DO 164 I=1,NFF
      DO 164 J=1,NFF
      II=I+NFB
      JJ=J+NFB
  164 S(II,JJ)=X(I,J)
      RETURN
      END
```

Figure 97. Subroutine COSTAT Program Listing

```
      SUBROUTINE TRANS(AMT,X,T,DUD,NX,MX,NF,MU,MF,B,IF)
      DIMENSION AMT(MF,MX),X(MX,MX),T(MF,MF),DUD(MU,MU),B(MF,MX),IF(MF)
      DO 1 I=1,NF
      DO 1 J=1,NX
      B(I,J)=0.
      DO 1 K=1,NX
    1 B(I,J)=B(I,J)+AMT(I,K)*X(K,J)
      DO 2 I=1,NF
      DO 2 J=1,NF
      T(I,J)=0.
      L=IF(I)
      M=IF(J)
      DO 2 K=1,NX
    2 T(I,J)=T(I,J)+DUD(L,M)*B(I,K)*AMT(J,K)
      RETURN
      END
```

Figure 98. Subroutine TRANS Program Listing

```
      SUBROUTINE UNSCR(T,DJDK,DJV,DJVT,IF,JF,NF,NU,NM,MU,MM,MF)
      DIMENSION T(MF,MF),DJDK(MU,MM),DJV(MF),DJVT(MF)
      DIMENSION IF(MF),JF(MF)
      L=1
      DO 1 I=1,NU
      DO 1 J=1,NM
      IF(L.GT.NF) GO TO 1
      IF(I.NE.IF(L)) GO TO 1
      IF(J.NE.JF(L)) GO TO 1
      DJV(L)=DJDK(I,J)
      L=L+1
1     CONTINUE
      DO 3 I=1,NF
      DJVT(I)=0.
      DO 3 K=1,NF
3     DJVT(I)=DJVT(I)+T(I,K)*DJV(K)
      L=1
      DO 4 I=1,NU
      DO 4 J=1,NM
      IF(L.GT.NF) GO TO 5
      IF(I.NE.IF(L)) GO TO 5
      IF(J.NE.JF(L))  GO TO 5
      DJDK(I,J)=DJVT(L)
      L=L+1
      GO TO 4
5     DJDK(I,J)=0.
4     CONTINUE
      RETURN
      END
```

Figure 99. Subroutine UNSCR Program Listing

```
      SUBROUTINE GCAL(A,B,X,N,M,NN,MM,IMAX,E,ES,V,U,KWA,MX,IERR)
C
C THIS SUBROUTINE SOLVES THE GENERAL MATRIX EQUATION XA+BX=C
C FORM (I-A),(I+A),(I-B),AND (I+B) THEN INVERT (I-A) AND (I-B)
      DIMENSION A(MX,MX),B(MX,MX),X(MX,MX),E(NN,NN)
      DIMENSION ES(MX,MX),V(NN,NN),U(MM,MM),KWA(MX)
      EE=.0 1
      IERR=
      DO 1 I=1,N
      DO 2 J=1,N
    2 E(I,J) = -A(I,J)
    1 E(I,I)=E(I,I)+1.
      DO 3 I=1,M
      DO 4 J=1,M
    4 ES(I,J) = -B(I,J)
    3 ES(I,I)=ES(I,I)+1.
      NRR=N
      NCC=N
      NR=NN
      CALL TDINVR(ISOL,IDSOL,NRR,NCC,E,NR,KWA,DET)
      IF((ISOL+IDSOL).LE.2) GO TO 5
      IERR=4
      GO TO 601
    5 NR=MX
      CALL TDINVR(ISOL,IDSOL,M,M,ES,NR,KWA,DET)
      IF((ISOL+IDSOL).LE.2) GO TO 6
      IERR=
      GO TO 601
    6 NC=N*
C
C FORM U,V,AND W
C
      DO 11 I=1,N
      DO 11 J=1,N
      V(I,J) = E(I,J)
      DO 11 K=1,N
   11 V(I,J) = V(I,J) + A(I,K)*E(K,J)
      DO 12 I=1,M
      DO 12 J=1,M
      U(I,J) = ES(I,J)
      DO 12 K=1,M
   12 U(I,J) = U(I,J)+ES(I,K)*B(K,J)
      DO 13 I=1,M
      DO 13 J=1,N
      B(I,J)=0.
      DO 14 K=1,M
   14 B(I,J) = B(I,J) + ES(I,K) * A(K,J)
   13 B(I,J)=2.*B(I,J)
      DO 15 I=1,M
      DO 15 J=1,N
      X(I,J) = 0.
      DO 15 K=1,N
   15 X(I,J)= X(I,J) + B(I,K) * E(K,J)
      ITER=
  100 CONTINUE
      DO 30 I=1,M
      DO 30 J=1,N
      B(I,J)=0.
      DO 30 K=1,M
   30 B(I,J) = B(I,J) + U(I,K)*X(K,J)
C CONVERGENCE CHECK
C
      ICOT=
```

Figure 100. Subroutine GCAL Program Listing

```
      DO 31 I=1,M
      DO 31 J=1,N
      DX=0.
      DO 32 K=1,N
   32 DX = DX + B(I,K)*V(K,J)
      X(I,J) = X(I,J) + DX
      AX=ABS(X(I,J))
      IF(AX.LT.1.E-20) GO TO 42
      IF(AX.LT.1.E 20) GO TO 41
      IFRR=1
      GO TO 601
   41 RAT=ABS(DX/X(I,J))
      IF(RAT-EE)42,42,43
   42 ICOT=ICOT+1
   43 CONTINUE
   31 CONTINUE
      ITER=ITER+1
      IF(ICOT-NC)44,50,44
   44 CONTINUE
      DO 33 I=1,N
      DO 33 J=1,N
   33 E(I,J)=V(I,J)
      DO 34 I=1,M
      DO 34 J=1,M
   34 ES(I,J)=U(I,J)
      DO 45 I=1,N
      DO 45 J=1,N
      V(I,J)=0.
      DO 45 K=1,N
   45 V(I,J) = V(I,J) + E(I,K)*E(K,J)
      DO 46 I=1,M
      DO 46 J=1,M
      U(I,J)=0.
      DO 46 K=1,M
   46 U(I,J) = U(I,J) + ES(I,K)*ES(K,J)
      IF(ITER.LT.IMAX) GO TO 100
      WRITE(9,600)
  600 FORMAT(/7X,12H ITER = IMAX)
   50 CONTINUE
      RETURN
  601 WRITE(9,602) IFRR
  602 FORMAT(/7X,7H IFRR =,I2)
      RETURN
      END
```

Figure 100. Subroutine GCAL Program Listing (Concluded)

```
      SUBROUTINE CAL(A,XN,P,KWA,N,NR,IMAX,IT,IERR)
      DIMENSION A(NR,1),XN(NR,1),P(NR,1),KWA(NR)
      IERR=
      TR=0.
      DO 30  I=1,N
  300 TR=TR+A(I,I)
      FN=N
      IF(TR)301,2,2
2     IERR=1
      GO TO 661
  301 ALF=ABS(TR)/FN
      EE=.001
      NC=N*(N+1)
      NC=NC/2
      DO 60 I=1,N
      DO 63 J=1,N
      GOTO(61,62),IT
   61 P(I,J)=A(I,J)
      GOTO 63
   62 P(I,J)=A(J,I)
   63 CONTINUE
      P(I,I)=P(I,I)-ALF
   60 CONTINUE
      CALL TDINVR(ISOL,IDSOL,N,N,P,NR,KWA,DET)
      IF((ISOL+IDSOL).LE.2) GO TO 22
      IERR=3
      GO TO 661
   22 DO 4 I=1,N
      DO 4 J=1,N
      A(I,J)=0.
      DO 4 K=1,N
    4 A(I,J)=A(I,J)+P(K,I)*XN(K,J)*2.*ALF
      DO 5 I=1,N
      DO 5 J=1,N
      XN(I,J)=0.
      DO 5 K=1,N
    5 XN(I,J)=XN(I,J)+A(I,K)*P(K,J)
      DO 7 I=1,N
      DO 8 J=1,N
    8 P(I,J)=P(I,J)*2.*ALF
    7 P(I,I)=P(I,I)+1.
      ITER=
  100 CONTINUE
      DO 9 I=1,N
      DO 9 J=1,N
      A(I,J)=0.
      DO 9 K=1,N
    9 A(I,J)=A(I,J)+P(K,I)*XN(K,J)
      ICOT=
      DO 10 I=1,N
      DO 10 J=I,N
      DXIJ= .
      DO 11 K=1,N
   11 DXIJ=DXIJ+A(I,K)*P(K,J)
      XN(I,J)=XN(I,J)+DXIJ
      XN(J,I)=XN(I,J)
      AXN=ABS(XN(I,J))
      IF(AXN.LT.1.E-20) GO TO 14
      IF(AXN.LT.1.E 20) GO TO 201
      IERR=2
      GO TO 661
  201 RAT=ABS(DXIJ/XN(I,J))
      IF(RAT-EE)14,14,70
```

Figure 101. Subroutine CAL Program Listing

```
 14 ICOT=ICOT+1
 70 CONTINUE
 10 CONTINUE
 18 ITER=ITER+1
    IF(ICOT-NC)15,50,15
 15 CONTINUE
    DO 20 I=1,N
    DO 20 J=1,N
 20 A(I,J)=P(I,J)
 16 DO 17 I=1,N
    DO 17 J=1,N
    P(I,J)=0.
    DO 17 K=1,N
 17 P(I,J)=P(I,J)+A(I,K)*A(K,J)
 40 IF(ITER.LT.IMAX) GO TO 100
    WRITE(9,600)
600 FORMAT(/7X,12H ITER = IMAX)
 50 CONTINUE
    RETURN
601 WRITE(9,602) IERR
602 FORMAT(/7X,6H IERR=I2)
    RETURN
    END
```

Figure 101. Subroutine CAL Program Listing (Concluded)

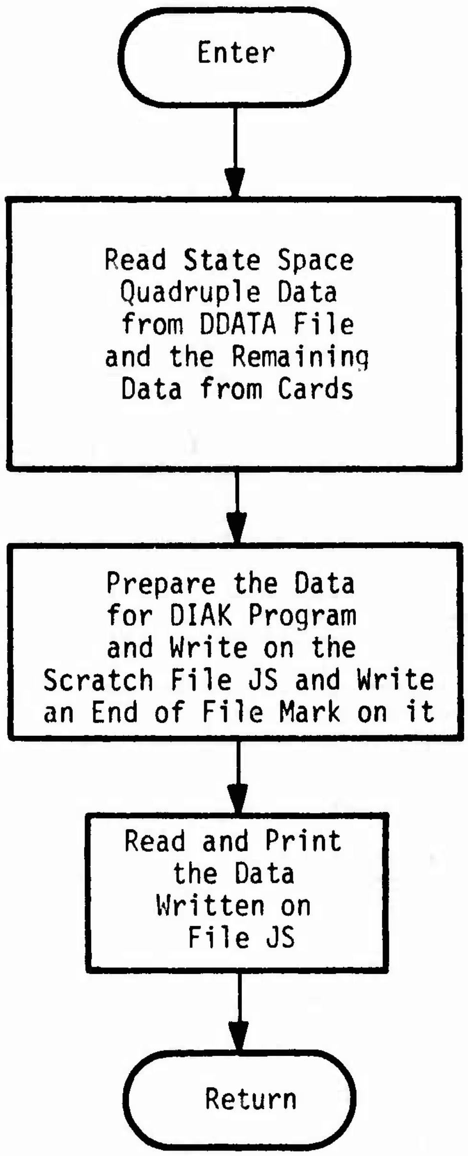

Figure 102. Subroutine DDIAK Flow Chart

```
      SUBROUTINE DDIAK(A,B,C,D,B1,B2,C1,C3,D11,BK,NXM,NRM,NUM)
C
C     PURPOSE - TO PREPARE DATA FOR DIAK PROGRAM
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        ZERO
C        FILE
C        MPRS
C        ERRM
C        WTP
C        INPTM
C
C     ARGUMENTS LIST
C        A                         STATE TRANSITION MATRIX
C        B                         CONTROL INPUT MATRIX
C        C                         STATE OUTPUT MATRIX
C        D                         CONTROL OUTPUT MATRIX
C        B1                        INPUT MATRIX FOR CONTROL INPUTS - G1
C        B2                        INPUT MATRIX FOR GUST INPUTS - G2
C        C1                        STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H
C        C3                        STATE OUTPUT MATRIX FOR MEASUREMENTS - M
C        D11                       OUTPUT MATRIX FOR DESIGN OUTPUTS - D
C        BK                        FEEDBACK GAIN MATRIX
C        NXM          INPUT        MAXIMUM NO OF STATES
C        NRM          INPUT        MAXIMUM NO OF OUTPUTS
C        NUM          INPUT        MAXIMUM NO OF INPUTS
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION B1(NXM,NUM),B2(NXM,NUM)
      DIMENSION C1(NRM,NXM),C3(NRM,NXM)
      DIMENSION D11(NRM,NUM),BK(NUM,NRM)
      DIMENSION HEAD(20),CARD(20)
      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20)
     1,JQ,JS,JSD,JE,JO
      DATA HBFBB,HBG1B,HBG2B,HBHBB/4H F   ,4H G1 ,4H G2 ,4H H  /
      DATA HBDBB,HBAMB,HC,HBCAR,HREAD/4H D   ,4H AM ,1HC,4H CAR,4HREAD/
      DATA HBTAP,HEND/4H TAP,4HEND /
      DATA HBBBB,HPEBB/4H    ,4HPE  /
      DATA HBBKB/4H BK /
C
C     READ IF DATA IS ON CARDS ONLY
C
      READ(IR,20)CARD
   20 FORMAT(20A4)
      IF(CARD(6).EQ.HBHBB)GO TO 83
      IF(CARD(6).NE.HPEBB)GO TO 162
      CALL ZERO(A,NXM,NXM)
      CALL ZERO(B,NXM,NUM)
      CALL ZERO(C,NRM,NUM)
      CALL ZERO(D,NRM,NUM)
      CALL ZERO(BK,NUM,NRM)
      READ(IR,20)HEAD
      CALL FILE(JO,LOCATE,HEAD)
      READ(JO)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),
     1((B(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),
     2((D(I,J),I=1,NR),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3
C
C     PARTITION MATRICES B,C,D
C
      IF(NU1.LE.0)STOP 111
      IF(NU2.LE.0)STOP 111
```

Figure 103. Subroutine DDIAK Program Listing

```
      IF(NR1.LE.0)STOP 111
      IF(NR3.LE.0)STOP 111
      DO 28 I=1,NX
      DO 24 J=1,NU1
   24 B1(I,J)=B(I,J)
      DO 28 J=1,NU2
      JJ=NU1+J
   28 B2(I,J)=B(I,JJ)
      DO 40 J=1,NX
      DO 34 I=1,NR1
   34 C1(I,J)=C(I,J)
      DO 40 I=1,NR3
      II=NR1+NR2+I
   40 C3(I,J)=C(II,J)
      DO 44 I=1,NR1
      DO 44 J=1,NU1
   44 D11(I,J)=D(I,J)
      IF(IPRINT.LT.6) GO TO 80
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA    )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HB    )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC    )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD    )
      CALL MPRS(B1,NXM,NUM,NX,NU1,T,4HB1   )
      CALL MPRS(B2,NXM,NUM,NX,NU2,T,4HB2   )
      CALL MPRS(C1,NRM,NXM,NR1,NX,T,4HC1   )
      CALL MPRS(C3,NRM,NXM,NR3,NX,T,4HC3   )
      CALL MPRS(D11,NRM,NUM,NR1,NU1,T,4HD11 )
   80 CONTINUE
C
C     ORGANIZE CARD AND TAPE DATA ON TAPE
C
  100 READ(IR,120) CARD
  120 FORMAT(20A4)
      IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBTAP))GO TO 160
      IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBCAR))GO TO 100
      IF(CARD(1).EQ.HEND) GO TO 300
      WRITE(JS,120) CARD
      GO TO 100
  160 CONTINUE
      IF(CARD(6).EQ.HBFBB) GO TO 180
      IF(CARD(6).EQ.HBG1B) GO TO 200
      IF(CARD(6).EQ.HBG2B) GO TO 220
      IF(CARD(6).EQ.HBHBB) GO TO 240
      IF(CARD(6).EQ.HBDBB) GO TO 260
      IF(CARD(6).EQ.HBAMB) GO TO 280
      IF(CARD(6).EQ.HBBKB)GO TO 285
  162 CONTINUE
      WRITE(IW,165)
  165 FORMAT(//1X,24HINPUT CONTROL CARD ERROR)
      CALL ERRM(1,4HDDIA,4HK    ,3,0,IW)
C
C     WRITE MATRIX DATA ON SCRATCH FILE FOR DIAK PROGRAM
C
  180 CONTINUE
      CALL WTP(A,NX,NX,NXM,NXM,JS)
      GO TO 100
  200 CONTINUE
      CALL WTP(B1,NX,NU1,NXM,NUM,JS)
      GO TO 100
  220 CONTINUE
      CALL WTP(B2,NX,NU2,NXM,NUM,JS)
      GO TO 100
  240 CONTINUE
      CALL WTP(C1,NR1,NX,NRM,NXM,JS)
      GO TO 100
  260 CONTINUE
```

Figure 103. Subroutine DDIAK Program Listing (Continued)

```
      CALL WTP(D11,NR1,NU1,NRM,NUM,JS)
      GO TO 100
  280 CONTINUE
      CALL WTP(C3,NR3,NX,NRM,NXM,JS)
      GO TO 100
  285 CONTINUE
C
C     READ GAIN MATRIX FROM DDATA FILE
C
      READ(IR,20)HEAD
  290 CONTINUE
      READ(JD,20)CARD
      DO 295 I=1,20
      IF(CARD(I).NE.HEAD(I))GO TO 290
  295 CONTINUE
      CALL INPTM(BK,NUM,NRM,JD)
      REWIND JD
      CALL WTP(BK,NU1,NR3,NUM,NRM,JS)
      GO TO 100
  300 CONTINUE
      END FILE JS
      REWIND JS
      IF((IPRINT.LT.5).AND.(IPRINT.NE.3)) GO TO 400
C
C     READ AND PRINT OUT TAPE
C
      WRITE(IW,310)
  310 FORMAT(1H1,1X,23H*** DIAK INPUT DATA ***,//)
  320 CONTINUE
      READ(JS,120)CARD
      IF (EOF(JS)) 360,340
  340 WRITE(IW,350) CARD
  350 FORMAT(1X,20A4)
      GO TO 320
  360 CONTINUE
      REWIND JS
      WRITE(IW,380)
  380 FORMAT(//,1X,30H*** END OF DIAK INPUT DATA ***)
  400 CONTINUE
      RETURN
      END
```

Figure 103. Subroutine DDIAK Program Listing (Concluded)

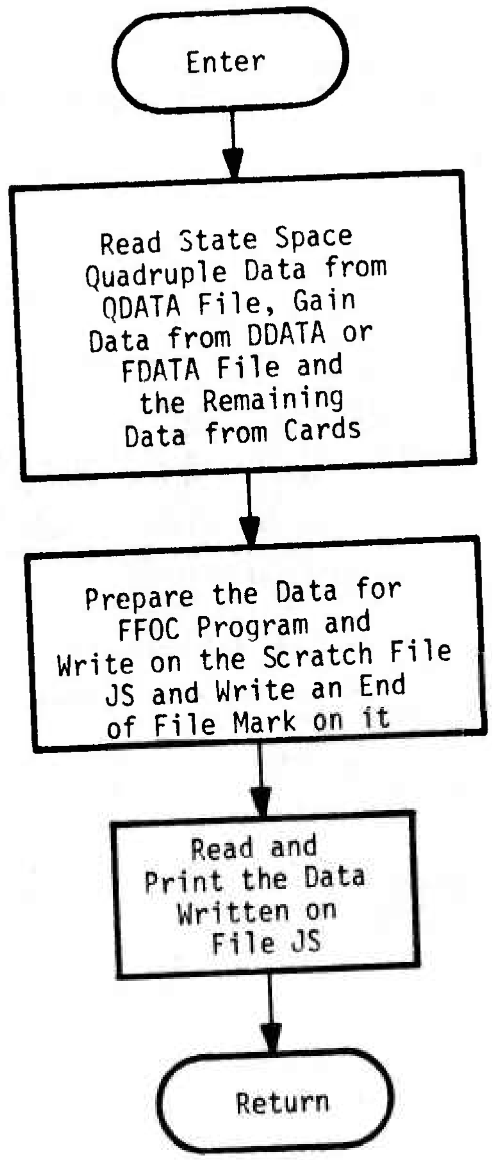

Figure 104. Subroutine DFFOC Flow Chart

```
      SUBROUTINE DFFOC(A,B,C,D,B1,B2,C1,C3,D11,BK,NXM,NRM,NUM)
C
C     PURPOSE - TO PREPARE DATA FOR FFOC PROGRAM
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        ZERO
C        FILE
C        MPRS
C        WTP
C        INPTM
C
C     ARGUMENTS LIST
C        A                      STATE TRANSITION MATRIX
C        B                      CONTROL INPUT MATRIX
C        C                      STATE OUTPUT MATRIX
C        D                      CONTROL OUTPUT MATRIX
C        B1                     INPUT MATRIX FOR CONTROL INPUTS - G1
C        B2                     INPUT MATRIX FOR GUST INPUTS - G2
C        C1                     STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H
C        C3                     STATE OUTPUT MATRIX FOR MEASUREMENTS - M
C        D11                    OUTPUT MATRIX FOR DESIGN OUTPUTS - D
C        BK                     FEEDBACK GAIN MATRIX
C        NXM        INPUT       MAXIMUM NO OF STATES
C        NRM        INPUT       MAXIMUM NO OF OUTPUTS
C        NUM        INPUT       MAXIMUM NO OF INPUTS
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION B1(NXM,NUM),B2(NXM,NUM)
      DIMENSION C1(NRM,NXM),C3(NRM,NXM)
      DIMENSION D11(NRM,NUM),BK(NUM,NRM)
      DIMENSION HEAD(20),CARD(20)
      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20)
     1,JO,JS,JSD,JF,JD
      DATA HBFBB,HBG1B,HBG2B,HBHBB/4H F  ,4H G1 ,4H G2 ,4H H  /
      DATA HBDBB,HBAMB,HC,HBCAR,HREAD/4H D  ,4H AM ,1HC,4H CAR,4HREAD/
      DATA HBTAP,HEND/4H TAP,4HEND /
      DATA HBBBB,HPERB/4H    ,4HPE  /
      DATA HBAKG,HBAKP,HBDEL/4H AKG,4H AK(,4H DEL/
C
C     READ IF DATA IS ON CARDS ONLY
C
      READ(IR,20)CARD
      IF(CARD(6).EQ.HBBBB) GO TO 80
      IF(CARD(6).NE.HPERB) GO TO 162
      CALL ZERO(A,NXM,NXM)
      CALL ZERO(B,NXM,NUM)
      CALL ZERO(C,NRM,NXM)
      CALL ZERO(D,NRM,NUM)
      CALL ZERO(BK,NUM,NRM)
      READ(IR,20)HEAD
   20 FORMAT(20A4)
      CALL FILE(JO,LOCATE,HEAD)
      READ(JO)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),
     1((B(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),
     2((D(I,J),I=1,NR),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3
C
C     PARTITION MATRICES B,C,D
C
      IF(NU1.LE.0)STOP 111
      IF(NU2.LE.0)STOP 111
      IF(NR1.LE.0)STOP 111
```

Figure 105. Subroutine DFFOC Program Listing

```
      IF(NR3.LE.0)STOP 111
      DO 28 I=1,NX
      DO 24 J=1,NU1
   24 B1(I,J)=B(I,J)
      DO 28 J=1,NU2
      JJ=NU1+J
   28 B2(I,J)=B(I,JJ)
      DO 40 J=1,NX
      DO 34 I=1,NR1
   34 C1(I,J)=C(I,J)
      DO 40 I=1,NR3
      II=NR1+NR2+I
   40 C3(I,J)=C(II,J)
      DO 44 I=1,NR1
      DO 44 J=1,NU1
   44 D11(I,J)=D(I,J)
      IF(IPRINT.LT.6) GO TO 80
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA    )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HB    )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC    )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD    )
      CALL MPRS(B1,NXM,NUM,NX,NU1,T,4HB1   )
      CALL MPRS(B2,NXM,NUM,NX,NU2,T,4HB2   )
      CALL MPRS(C1,NRM,NXM,NR1,NX,T,4HC1   )
      CALL MPRS(C3,NRM,NXM,NR3,NX,T,4HC3   )
      CALL MPRS(D11,NRM,NUM,NR1,NU1,T,4HD11 )
   80 CONTINUE
C
C     ORGANIZE CARD AND TAPE DATA ON TAPE
C
  100 READ(IR,120) CARD
  120 FORMAT(20A4)
      IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBTAP))GO TO 160
      IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBCAR))GO TO 100
      IF(CARD(1).EQ.HEND) GO TO 300
      WRITE(JS,120) CARD
      GO TO 100
  160 CONTINUE
      IF(CARD(6).EQ.HBFBB) GO TO 180
      IF(CARD(6).EQ.HBG1B) GO TO 200
      IF(CARD(6).EQ.HBG2B) GO TO 220
      IF(CARD(6).EQ.HBHBB) GO TO 240
      IF(CARD(6).EQ.HBDBB) GO TO 260
      IF(CARD(6).EQ.HBAMB) GO TO 280
      IF(CARD(6).EQ.HBAKG)GO TO 285
      IF(CARD(6).EQ.HBAKP)GO TO 285
      IF(CARD(6).EQ.HBDEL)GO TO 285
  162 CONTINUE
      WRITE(IW,165)
  165 FORMAT(//1X,24HINPUT CONTROL CARD ERROR)
      STOP 111
C
C     WRITE MATRIX DATA ON SCRATCH FILE FOR FFOC PROGRAM
C
  180 CONTINUE
      CALL WTP(A,NX,NX,NXM,NXM,JS)
      GO TO 100
  200 CONTINUE
      CALL WTP(B1,NX,NU1,NXM,NUM,JS)
      GO TO 100
  220 CONTINUE
      CALL WTP(B2,NX,NU2,NXM,NUM,JS)
      GO TO 100
  240 CONTINUE
      CALL WTP(C1,NR1,NX,NRM,NXM,JS)
      GO TO 100
```

Figure 105. Subroutine DFFOC Program Listing (Continued)

```
  260 CONTINUE
      CALL WTP(D11,NR1,NU1,NRM,NUM,JS)
      GO TO 100
  280 CONTINUE
      CALL WTP(C3,NR3,NX,NRM,NUM,JS)
      GO TO 100
C
C     READ GAINS FROM DDATA OR FDATA FILE
C
  285 CONTINUE
      JDF=JF
      IF(CARD(6).EQ.HBAKG) JDF=JD
      READ(IR,120)HEAD
  290 CONTINUE
      READ(JDF,120)CARD
      DO 295 I=1,20
      IF(CARD(I).NE.HEAD(I))GO TO 290
  295 CONTINUE
      CALL ZERO(BK,NUM,NRM)
      CALL INPTM(BK,NUM,NRM,JDF)
      REWIND JDF
      CALL WTP(BK,NU1,NR3,NUM,NRM,JS)
      GO TO 100
  300 CONTINUE
      END FILE JS
      REWIND JS
      IF((IPRINT.LT.5).AND.(IPRINT.NE.3)) GO TO 400
C
C     READ AND PRINT OUT TAPE
C
      WRITE(IW,310)
  310 FORMAT(1H1,1X,23H*** FFOC INPUT DATA ***,//)
  320 CONTINUE
      READ(JS,120)CARD
      IF (EOF(JS)) 360,340
  340 WRITE(IW,350) CARD
  350 FORMAT(//1X,20A4)
      GO TO 320
  360 CONTINUE
      REWIND JS
  400 CONTINUE
      RETURN
      END
```

Figure 105. Subroutine DFFOC Program Listing (Concluded)

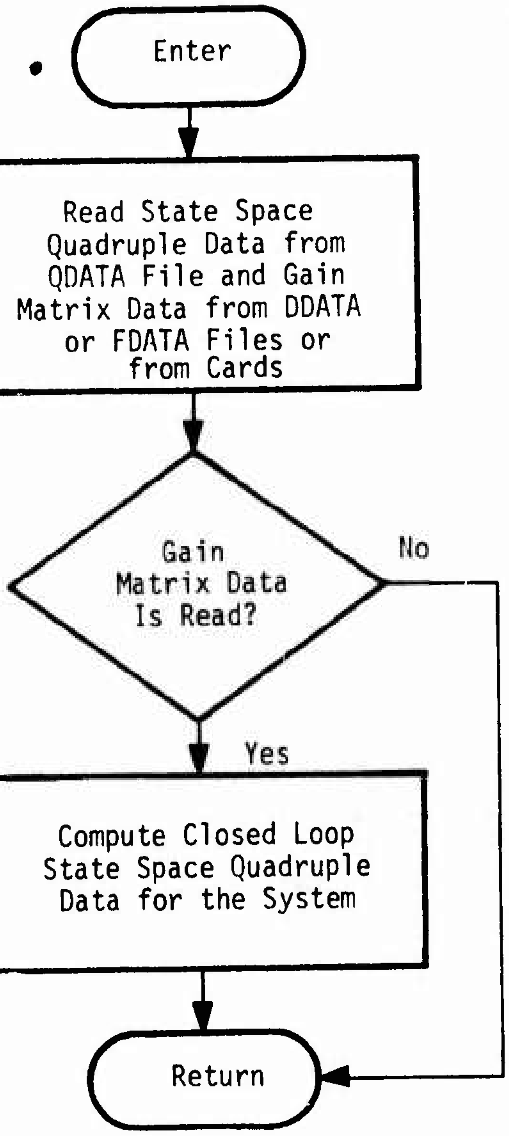

Figure 106. Subroutine DLSA Flow Chart

```
      SUBROUTINE DLSA(A,B,C,D,B1,B2,C1,C3,D11,BK,BKC3,NX,NR,NU,
     1NXM,NRM,NUM)
C
C     PURPOSE - TO PREPARE DATA FOR LSA PROGRAM
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        ZERO
C        FILE
C        MPRS
C        INPTM
C
C     ARGUMENTS LIST
C        A                         STATE TRANSITION MATRIX
C        B                         CONTROL INPUT MATRIX
C        C                         STATE OUTPUT MATRIX
C        D                         CONTROL OUTPUT MATRIX
C        B1                        INPUT MATRIX FOR CONTROL INPUTS - G1
C        B2                        INPUT MATRIX FOR GUST INPUTS - G2
C        C1                        STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H
C        C3                        STATE OUTPUT MATRIX FOR MEASUREMENTS - M
C        D11                       OUTPUT MATRIX FOR DESIGN OUTPUTS - D
C        BK                        FEEDBACK GAIN MATRIX
C        BKC3                      BK*C3
C        NXM        INPUT          MAXIMUM NO OF STATES
C        NRM        INPUT          MAXIMUM NO OF OUTPUTS
C        NUM        INPUT          MAXIMUM NO OF INPUTS
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION B1(NXM,NUM),B2(NXM,NUM)
      DIMENSION C1(NRM,NXM),C3(NRM,NXM)
      DIMENSION D11(NRM,NUM),BK(NUM,NRM)
      DIMENSION BKC3(NUM,NXM)
      DIMENSION HEAD(20),CARD(20)
      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20)
     1,JQ,JS,JSD,JF,JD
      DATA HBFBB,HBG1B,HBG2B,HBHBB/4H F  ,4H G1 ,4H G2 ,4H H  /
      DATA HBDBB,HBAKG,HC,HBCAR,HREAD/4H D  ,4H AKG,1HC,4H CAR,4HREAD/
      DATA HBTAP,HEND/4H TAP,4HEND /
      DATA HBBBB,HPEBB/4H    ,4HPE  /
      DATA HBAKP/4H AKP/
      IGAIN=0
C
C     READ QUADRUPLE DATA FROM QDATA FILE
C
      READ(IR,20) CARD
      IF(CARD(6).NE.HPEBB) GO TO 420
      CALL ZERO(A,NXM,NXM)
      CALL ZERO(B,NXM,NUM)
      CALL ZERO(C,NRM,NXM)
      CALL ZERO(D,NRM,NUM)
      CALL ZERO(BK,NUM,NRM)
      READ(IR,20) HEAD
   20 FORMAT(20A4)
      CALL FILE(JQ,LOCATE,HEAD)
      READ(JQ)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),
     1((B(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),
     2((D(I,J),I=1,NR),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3
C
C     PARTITION MATRICES B,C,D
C
      DO 28 I=1,NX
```

Figure 107. Subroutine DLSA Program Listing

```
      DO 24 J=1,NU1
   24 B1(I,J)=B(I,J)
      DO 28 J=1,NU2
      JJ=NU1+J
   28 B2(I,J)=B(I,JJ)
      DO 40 J=1,NX
      DO 34 I=1,NR1
   34 C1(I,J)=C(I,J)
      DO 40 I=1,NR3
      II=NR1+NR2+I
   40 C3(I,J)=C(II,J)
      DO 44 I=1,NR1
      DO 44 J=1,NU1
   44 D11(I,J)=D(I,J)
      IF(IPRINT.LT.6) GO TO 400
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA    )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HB    )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC    )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD    )
      CALL MPRS(B1,NXM,NUM,NX,NU1,T,4HB1   )
      CALL MPRS(B2,NXM,NUM,NX,NU2,T,4HB2   )
      CALL MPRS(C1,NRM,NXM,NR1,NX,T,4HC1   )
      CALL MPRS(C3,NRM,NXM,NR3,NX,T,4HC3   )
      CALL MPRS(D11,NRM,NUM,NR1,NU1,T,4HD11 )
C
C     READ GAIN MATRIX DATA FROM DDATA OR FDATA FILE OR FROM INPUT DATA
C
  400 CONTINUE
      READ(IR,20) CARD
      IF(CARD(1).EQ.HEND) GO TO 600
      IGAIN=1
      IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBTAP)) GO TO 460
      IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBCAR)) GO TO 560
  420 CONTINUE
      WRITE(IW,440)
  440 FORMAT(//,1X,24HINPUT CONTROL CARD ERROR)
      STOP 111
  460 CONTINUE
      IGAIN=1
      IF(CARD(6).EQ.HBAKG) GO TO 480
      IF(CARD(6).EQ.HBAKP) GO TO 480
      GO TO 420
  480 CONTINUE
      JDF=JF
      IF(CARD(6).EQ.HBAKG)JDF=JD
      READ(IR,20)HEAD
  500 CONTINUE
      READ(JDF,20)CARD
      DO 520 I=1,20
      IF(CARD(I).NE.HEAD(I))GO TO 500
  520 CONTINUE
      CALL ZERO(BK,NUM,NRM)
      CALL INPTM(BK,NUM,NRM,JDF)
      REWIND JDF
      GO TO 400
  560 CONTINUE
      CALL ZERO(BK,NUM,NRM)
      CALL INPTM(BK,NUM,NRM,IR)
      GO TO 400
  600 CONTINUE
      IF(IGAIN.EQ.0)RETURN
C
C     COMPUTE CLOSED LOOP QUADRUPLE DATA
C
      DO 70 I=1,NU1
      DO 70 J=1,NX
```

Figure 107. Subroutine DLSA Program Listing (Continued)

```
      BKC3(I,J)=0.0
      DO 70 K=1,NR3
   70 BKC3(I,J)=BKC3(I,J)+BK(I,K)*C3(K,J)
      DO 80 I=1,NX
      DO 80 J=1,NX
      DO 80 K=1,NU1
   80 A(I,J)=A(I,J)+B1(I,K)*BKC3(K,J)
      DO 90 I=1,NR1
      DO 90 J=1,NX
      DO 90 K=1,NU1
   90 C(I,J)=C(I,J)+D11(I,K)*BKC3(K,J)
      DO 100 I=1,NX
      DO 100 J=1,NU2
  100 B(I,J)=B2(I,J)
      NU=NU2
      IF(IPRINT.LT.6) RETURN
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA    )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HB    )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC    )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD    )
      CALL MPRS(BK,NUM,NXM,NU1,NX,T,4HBK    )
      RETURN
      END
```

Figure 107. Subroutine DLSA Program Listing (Concluded)

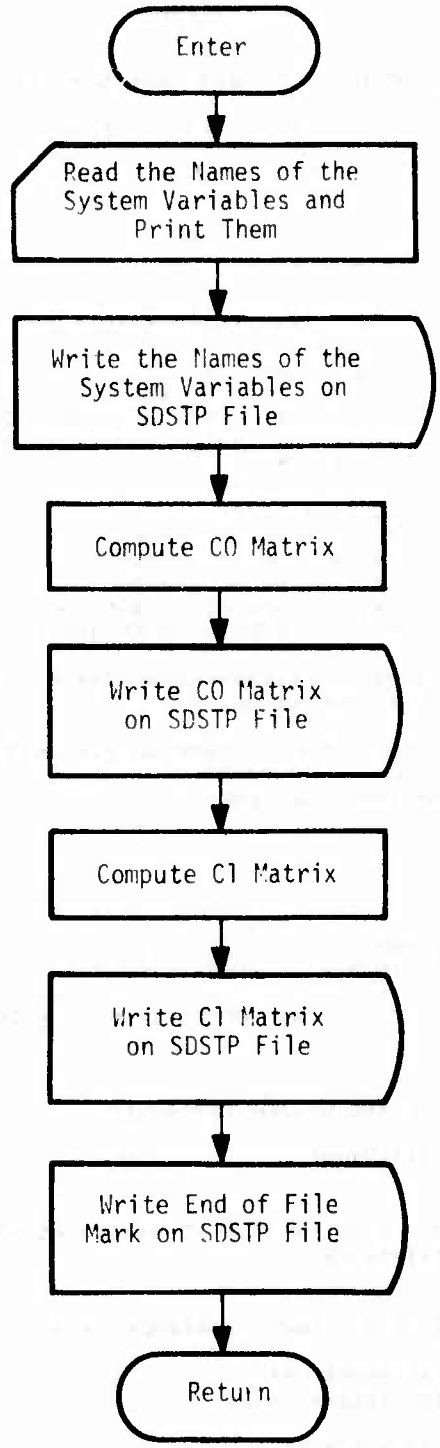

Figure 108. Subroutine FINK Flow Chart

```
      SUBROUTINE FINK(A,B,C,D,CC,NAME,NX,NR,NU,
     1NXM,NRM,NUM,NXRM,NXRUM)
C
C     PURPOSE - TO COMPUTE FREQUENCY DOMAIN REPRESENTATION
C     OF STATE SPACE QUADRUPLE DATA
C     ANALYSTS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     SUBPROGRAMS CALLED
C        ZERO
C        MPYS
C
C     ARGUMENTS LIST
C        A            INPUT      STATE TRANSITION MATRIX
C        B            INPUT      CONTROL INPUT MATRIX
C        C            INPUT      STATE OUTPUT MATRIX
C        D            INPUT      CONTROL OUTPUT MATRIX
C        CC                      FOR STORING CO AND CI MATRICES
C        NAME                    ARRAY FOR SYSTEM VARIABLES NAMES
C        NX           INPUT      NO OF STATES
C        NR           INPUT      NO OF OUTPUTS
C        NU           INPUT      NO OF INPUTS
C        NXM          INPUT      MAXIMUM NO OF STATES
C        NRM          INPUT      MAXIMUM NO OF OUTPUTS
C        NUM          INPUT      MAXIMUM NO OF INPUTS
C        NXRM         INPUT      MAXIMUM ROW DIMENSION FOR CO AND CI
C        NXRUM        INPUT      MAXIMUM COLUMN DIMENSION FOR CO AND CI
C
      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
      DIMENSION CC(NXRM,NXRUM),NAME(NXRUM)
      DIMENSION CARD(20)
      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),
     1JQ,JS,JSD,JF,JD
      DATA HC,HEND,HNAME/1HC,4HEND ,4HNAME/
      NXR=NX+NR
      NXRU=NXR+NU
  120 CONTINUE
      READ(IR,140)CARD
  140 FORMAT(20A4)
      IF(CARD(1).EQ.HEND)RETURN
      IF(CARD(1).EQ.HNAME)GO TO 200
      WRITE(IW,180)
  180 FORMAT(//,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)
      STOP 111
  200 CONTINUE
C
C     READ AND WRITE NAMES OF THE SYSTEM VARIABLES
C
      READ(IR,370)(NAME(I),I=1,NXRU)
  370 FORMAT(8A10)
      WRITE(IW,375)
  375 FORMAT(1H1,//,1X,29HNAMES OF THE OUTPUT VARIABLES,//)
      WRITE(IW,380)(NAME(I),I=1,NXR)
  380 FORMAT(1X,A10)
      WRITE(IW,385)
  385 FORMAT(//,1X,28HNAMES OF THE INPUT VARIABLES,//)
      NXRP1=NXR+1
      WRITE(IW,380)(NAME(I),I=NXRP1,NXRU)
      WRITE(JSD)NXRU,NU,(NAME(I),I=1,NXRU)
C
C     COMPUTE CC AND WRITE ON SDSTP FILE
C
      CALL ZERO(CC,NXRM,NXRUM)
```

Figure 109. Subroutine FINK Program Listing

```
      DO 30  I=1,NX
      DO 28  J=1,NX
  280 CC(I,J)=-A(I,J)
      DO 30  J=1,NU
      JJ=NX+J
  300 CC(I,JJ)=B(I,J)
      DO 34  I=1,NR
      II=NX+I
      DO 32  J=1,NX
  320 CC(II,J)=-C(I,J)
      DO 34  J=1,NU
      JJ=NX+J
  340 CC(II,JJ)=D(I,J)
      DO 36  I=1,NR
      II=NX+I
  360 CC(II,II)=1.0
      CALL MPRS(CC,NXRM,NXRUM,NXR,NXRU,T,4HC0  )
      WRITE(JSD)((CC(I,J),J=1,NXRU),I=1,NXR)
C
C     COMPUTE C1 AND WRITE ON SDSTP FILE
C
      CALL ZERO(CC,NXRM,NXRUM)
      DO 26  I=1,NX
  260 CC(I,I)=1.0
      CALL MPRS(CC,NXRM,NXRUM,NXR,NXRU,T,4HC1  )
      WRITE(JSD)((CC(I,J),J=1,NXRU),I=1,NXR)
C
C     WRITE AN END OF FILE MARK ON SDSTP
C
      ENDFILE JSD
      GO TO 120
      END
```

Figure 109. Subroutine FINK Program Listing (Concluded)

```
      SUBROUTINE MP(K,L,I,J,A)
      DIMENSION A(K,L)
      DO 1 II=1,I
      WRITE(9,5)II
    5 FORMAT(5H ROW I3)
    1 WRITE(9,2)(A(II,JJ),JJ=1,J)
2     FORMAT(2X,10E12.4)
      RETURN
      END
```

Figure 110. Subroutine MP Program Listing

```
      SUBROUTINE OUTP(I,J,II,JJ,Y,IR)
      DIMENSION Y(I,J),YD(5),ID(5),JD(5)
   50 FORMAT(5(2I2,F12.5))
      III=0
      DO 100 K=1,II
      DO 100 M=1,JJ
      IF(Y(K,M).EQ.0.) GOTO 100
      III=III+1
      YD(III)=Y(K,M)
      ID(III)=K
      JD(III)=M
      IF(III.LT.5)GOTO 100
      WRITE(IR,50)(ID(L),JD(L),YD(L),L=1,III)
      III=0
  100 CONTINUE
      IF(III.EQ.0) RETURN
      WRITE(IR,50)(ID(L),JD(L),YD(L),L=1,III)
      RETURN
      END
```

Figure 111. Subroutine OUTP Program Listing

```
      SUBROUTINE POLES(NX,A,MX,RR,M)
      DIMENSION A(MX,1),RR(1)
      CALL HESSEN(NX,A,MX)
      CALL QRCALL(MX,A,RR,M,NX)
      WRITE(9,6087)
 6087 FORMAT(1H1/7X,11HEIGENVALUES/12X,4HREAL,9X,9HIMAGINARY,8X,13HDAMPI
     1NG RATIO,5X,9HFREQUENCY//)
      MM=M/2
      DO 6083 K=1,MM
      I=2*K-1
      OMEGA=SQRT(RR(I)*RR(I)+RR(I+1)*RR(I+1))
      IF(ABS(RR(I+1)).GT..00000001) GO TO 1
      WRITE(9,6084) RR(I)
      GO TO 6083
 1    DELTA=RR(I)/OMEGA
      WRITE(9,6084) RR(I),RR(I+1),DELTA,OMEGA
 6083 CONTINUE
 6084 FORMAT(8X,4F15.8)
      RETURN
      END
```

Figure 112. Subroutine POLES Program Listing

```
      SUBROUTINE HESSEN(N,A,D)
      DIMENSION A(1)
      INTEGER P,PM,PX,D
      ID=D+1
      NN=(N-1)*ID+1
      KX=NN-ID-ID+1
      PM=1
      PX=N
      DO 70 K=2,KX,ID
      NK=PX
      PM=PM+D
      PX=PX+D
      JP=PM
      T=0.
      B=0.
      J=K
      JC=JP
      JK=J
   30 T=ABS(A(J))
      IF(T.LE.B) GO TO 35
      JC=JP
      JK=J
      B=T
   35 IF(J.GE.NK) GO TO 37
      J=J+1
      JP=JP+D
      GO TO 30
   37 IF(JK.EQ.K) GO TO 44
      J=JC
      DO 38 P=PM,PX
      T=A(P)
      A(P)=A(J)
      A(J)=T
   38 J=J+1
      P=JK
      DO 39 J=K,NN,D
      T=A(J)
      A(J)=A(P)
      A(P)=T
   39 P=P+D
   44 IF(A(K).EQ.0.) GO TO 70
      JC=PM+D
      JK=K+1
      T=1./A(K)
   45 B=A(JK)
      IF(B.EQ.0.) GO TO 65
      B=B*T
      KM=K+D
      JM=JK+D
   50 AJM=A(JM)-B*A(KM)
      IF(ABS(AJM).LE.(.1E-9*ABS(A(JM)))) AJM=0.
      A(JM)=AJM
      KM=KM+D
      JM=JM+D
      IF(JM.LE.NN) GO TO 50
      J=JC
      DO 60 P=PM,PX
      AP=A(P)+B*A(J)
      IF(ABS(AP).LE.(.1E-9*ABS(A(P)))) AP=0.
      A(P)=AP
   60 J=J+1
   65 JK=JK+1
      JC=JC+D
```

Figure 113. Subroutine HESSEN Program Listing

```
      IF(JK.LE.NK) GO TO 45
70    CONTINUE
      RETURN
      END
```

Figure 113. Subroutine HESSEN Program Listing (Concluded)

```
      SUBROUTINE QRCALL(D,A,R,M,NIN)
      INTEGER D
      DIMENSION A(D,1),R(1)
      N = NIN
      ANN = 1.
      ACT = .1E-7
      ITER = 0
      M = 0
      IF(N.LE.1)   RETURN
      IF(N.EQ.2)   GO TO 25
15    DELTA=ACT*ABS(A(N,N))
      ACC = ABS(A(N,N-1))
      IF(ACC.EQ.0.)   GO TO 16
      IF(ACC.GT.DELTA)   GO TO 25
      IF(ITER.GT.25)   GO TO 16
      IF(ANN.GT.ACT)    GO TO 25
16    M = M+2
      R(M-1)= A(N,N)
      R(M) = 0.
17    K = NIN-N+1
      ITER = 0
      N = N-1
20    IF(N.GT.2)   GO TO 15
      IF(N.EQ.2)   GO TO 25
      IF(N.EQ.1)   GO TO 16
      R(M+1)=ACT
      RETURN
25    B = .5*(A(N-1,N-1)+A(N,N))
      DAN=ABS(A(N,N)-A(N-1,N-1))
      SAN=ABS(A(N,N))+ABS(A(N-1,N-1))
      IF(DAN.LE.ACT*SAN)   DAN=0.
      DAN=DAN*DAN*.25
      C=A(N,N-1)*A(N-1,N)
      T=DAN+C
      IF( (C.LT.0.) .AND. (ABS(T).LE.ACT*DAN) )   T=0.
      IF(ABS(T).LE.ACT)   T=0.
      C = SQRT(ABS(T))
      IF(N.NE.2)   GO TO 50
26    IF(T.GE.0.)    GO TO 30
      M = M+2
      R(M-1) = B
      R(M) = C
27    N = N-1
      GO TO 17
30    M = M+2
      R(M-1) = B+C
      R(M) = 0.
      K = NIN-N+1
      M = M+2
      R(M-1) = B-C
      R(M) = 0.
      GO TO 27
50    IF(T.GE.0.)    GO TO 60
      R(M+5) = B
      R(M+6) = C
      R(M+7) = B
      R(M+8) = -C
      GO TO 70
60    X = B+C
      Y = B-C
      R(M+6) = 0.
      R(M+8) = 0.
      R(M+5) = X
```

Figure 114. Subroutine QRCALL Program Listing

```
      R(M+7) = Y
      IF(ABS(X).GT.ABS(Y))   GO TO 70
      R(M+5) = Y
      R(M+7) = X
70    IF(ITER.LE.0)   GO TO 130
      X = ABS(R(M+5)-R(M+1))+ABS(R(M+6)-R(M+2))
      ACC = ABS(R(M+5))+ABS(R(M+1))+ABS(R(M+6))+ABS(R(M+2))
      IF(ACC.GT.1.)   X=X/ACC
      Y = ABS(R(M+7)-R(M+3))+ABS(R(M+8)-R(M+4))
      ACC = ABS(R(M+7))+ABS(R(M+3))+ABS(R(M+8))+ABS(R(M+4))
      IF(ACC.GT.1.)   Y=Y/ACC
      ACC = ABS(A(N-1,N-2))
      DELTA=AMAX1(DELTA,(ACT*ABS(A(N-1,N-1))))
      IF(ACC.GT.DELTA)   GO TO 80
      IF(ITER.GT.25)   GO TO 26
      IF((X.LE.ACT).AND.(Y.LE.ACT))   GO TO 26
80    IF(ITER.GT.200)   GO TO 200
      IF((X.GT..5  ).AND.(Y.GT..5  ))   GO TO 130
      K = M+5
      IF(Y.GT..5  )   GO TO 120
      IF(X.GT..5  )   GO TO 110
      RHO = R(M+5)*R(M+7)-R(M+6)*R(M+8)
      SIGMA = R(M+5)+R(M+7)
100   CONTINUE
      ANN = A(N,N)
      CALL QR(N,A,RHO,SIGMA,D,DELTA)
      B = ABS(A(N,N))
      ANN = ABS(ANN-A(N,N))
      IF(B.GT.ACT)   ANN = ANN/B
      ITER = ITER+1
      DO 105 I=1,4
      K = M+I
105   R(K) = R(K+4)
      GO TO 15
110   K = M+7
120   RHO = R(K)*R(K)
      SIGMA = R(K)+R(K)
      GO TO 100
130   RHO = 0.
      SIGMA=0.
      GO TO 100
200   CONTINUE
      WRITE(9,700)
700   FORMAT(1H1,25HALL EIGENVALUES NOT FOUND)
      RETURN
      END
```

Figure 114. Subroutine QRCALL Program Listing (Concluded)

```
      SUBROUTINE QR(N,A,RHO,SIGMA,D,DELTA)
      DIMENSION A(1)
      REAL KAPPA
      INTEGER P,Q,D
      EQUIVALENCE (P,Q)
      ID = D+1
      N0 = ID*(N-1)+1
      N1 = N0-ID
      N2 = N1-ID
      N3 = N2-ID
      IF(N.GT.3) GO TO 5
      IF(N.LE.2) RETURN
2     Q = 1
      GO TO 35
5     I = N3+1
7     IF(ABS(A(I)).LT.DELTA) GO TO 10
      IF(I.LE.2) GO TO 2
      I = I-ID
      GO TO 7
10    Q = I+D
      A(I) = 0.
35    I = P
      IQ = 0
      I0 = I-D
      I1 = I+D
      I2 = I1+D
      G1 = A(I)*(A(I)-SIGMA)+A(I1)*A(I+1)+RHO
      G2 = A(I+1)*(A(I)+A(I1+1)-SIGMA)
      G3 = A(I+1)*A(I1+2)
      A(I+2) = 0.
      GO TO 45
40    G1 = A(I0)
      G2 = A(I0+1)
      G3 = 0.
      IQ = I0+D
      IF(I.LE.N2) G3 = A(I0+2)
45    KAPPA = SQRT(G1*G1+G2*G2+G3*G3)
      IF(G1.LT.0.) KAPPA = -KAPPA
      IF(KAPPA.NE.0.) GO TO 47
      ALPHA = 2.
      P1 = 0.
      P2 = 0.
      GO TO 48
47    ALPHA = 1.+G1/KAPPA
      P1 = 1./(G1+KAPPA)
      P2 = P1*G3
      P1 = P1*G2
48    IF(I.EQ.Q) GO TO 49
      A(I0) = -A(I0)
      IF(I.NE.P) A(I0) = -KAPPA
49    J = I-D
50    J = J+D
      IF(J.GE.N0) GO TO 51
      ETA = A(J)+P1*A(J+1)
      IF(I.LE.N2) ETA = ETA+P2*A(J+2)
      ETA = ALPHA*ETA
      A(J) = A(J)-ETA
      A(J+1) = A(J+1)-P1*ETA
      IF(I.LE.N2) A(J+2) = A(J+2)-P2*ETA
      GO TO 50
51    J = IQ-1
      JINX = MIN0(I+2,N1+1)
60    J = J+1
```

Figure 115. Subroutine QR Program Listing

```
      K = J+D
      ETA = A(J)+P1*A(K)
      L = K+D
      IF(I.LE.N2) ETA = ETA+P2*A(L)
      ETA = ETA*ALPHA
      A(J) = A(J)-ETA
      A(K) = A(K)-P1*ETA
      IF(I.LE.N2) A(L) = A(L)-P2*ETA
      IF(J.LT.JINX) GO TO 60
      IF(I.GT.N3) GO TO 65
      ETA = ALPHA*P2*A(I2+3)
      A(I+3) = -ETA
      A(I1+3) = -P1*ETA
      A(I2+3) = A(I2+3)-P2*ETA
65    IF(I.GE.N1) RETURN
      I0 = I +1
      I = I1+1
      I1 = I2+1
      I2 = I2+ID
      GO TO 4.
      END
```

Figure 115. Subroutine QR Program Listing (Concluded)

```
      SUBROUTINE INPTM(A,II,JJ,IR)
C
C     PURPOSE - TO READ NONZERO ELEMENTS OF A MATRIX FROM FILE IR
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        A          INPUT      MATRIX DATA
C        II         INPUT      MAXIMUM NO OF ROWS
C        JJ         INPUT      MAXIMUM NO OF COLUMNS
C        IR         INPUT      FILE NO FOR READING MATRIX DATA
C
      DIMENSION A(II,JJ),ID(5),JD(5),YD(5)
    2 FORMAT (5(2I2,E12.5))
    1 READ(IR,2)(ID(L),JD(L),YD(L),L=1,5)
      IF(EOF(IR))10,6
  6   CONTINUE
      IF(ID(1))3,10,3
    3 DO 5 L=1,5
      IF (ID(L))4,1,4
    4 I=ID(L)
      J=JD(L)
    5 A(I,J)=YD(L)
      GO TO 1
   10 CONTINUE
      RETURN
      END
```

Figure 116. Subroutine INPTM Program Listing

```
      SUBROUTINE WTP(A,NR,NC,NR4,NC4,JW)
C
C     PURPOSE - TO WRITE NONZERO ELEMENTS OF A MATRIX ON A FILE
C     ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     DATE WRITTEN - 1975
C
C     ARGUMENTS LIST
C        A           INPUT       MATRIX DATA
C        NR          INPUT       NO OF ROWS
C        NC          INPUT       NO OF COLUMNS
C        NR4         INPUT       MAXIMUM NO OF ROWS
C        NC4         INPUT       MAXIMUM NO OF COLUMNS
C        JW          INPUT       FILE NO FOR WRITING DATA
C
      DIMENSION A(NR4,NC4),BCARD(20)
      DIMENSION AD(5),ID(5),JD(5)
      INTEGER BCARD
      IF(NR.EQ.0)GO TO 100
      IF(NC.EQ.0)GO TO 100
      III=0
      DO 80 K=1,NR
      DO 80 M=1,NC
      IF(A(K,M).EQ.0.0)GO TO 80
      III=III+1
      AD(III)=A(K,M)
      ID(III)=K
      JD(III)=M
      IF(III.LT.5)GO TO 80
      WRITE(JW,60)(ID(L),JD(L),AD(L),L=1,III)
   60 FORMAT(5(2I2,E12.5))
      III=0
   80 CONTINUE
      IF(III.EQ.0)GO TO 100
      WRITE(JW,60)(ID(L),JD(L),AD(L),L=1,III)
  100 CONTINUE
      IBLANK=4H
      DO 110 I=1,20
  110 BCARD(I)=IBLANK
      WRITE(JW,120)(BCARD(I),I=1,20)
  120 FORMAT(20A4)
      RETURN
      END
```

Figure 117. Subroutine WTP Program Listing

```
      FUNCTION GRAN(N)
      X=N
      IF (N .EQ. 0) GO TO 1
      ISEED = 33973639892
      X=ISEED
      TEM=RANF(X)
      X=0.
    1 TEM = 0.0
      DO 2 I = 1,12
 2    TEM=TEM+RANF(X)
      TEM = TEM - 6.0
      GRAN = TEM
      RETURN
      END
```

Figure 118. Subroutine GRAN Program Listing

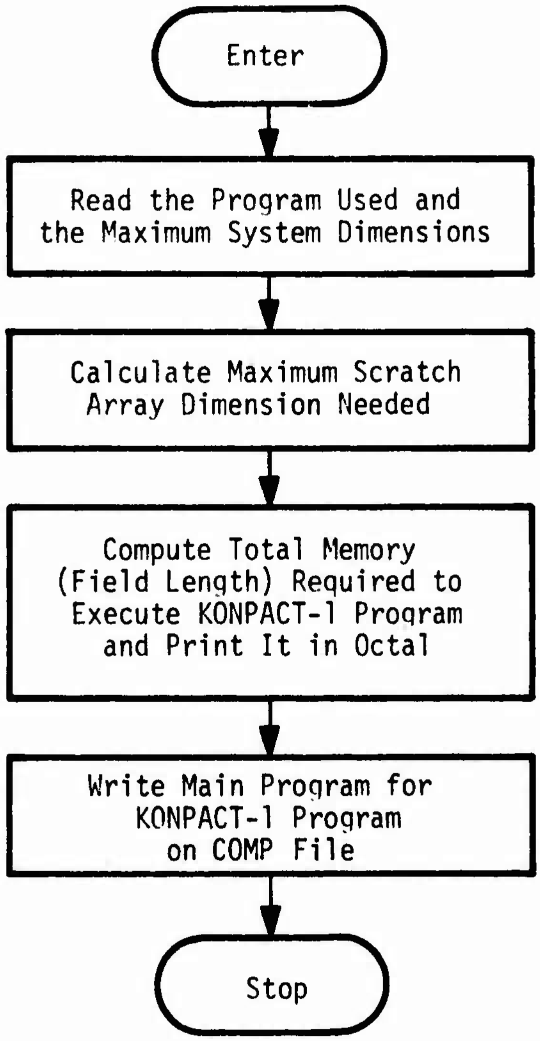

Figure 119. Program PRECOM Flow Chart

APPENDIX

PRECOMPILER PROGRAM FOR KONPACT-1

The precompiler program performs the task of writing the MAIN program for KONPACT-1. A brief description of the precompiler program is presented in this section.

The precompiler program reads the system dimensions and the KONPACT-1 program names and computes the maximum sizes of the scratch arrays. It writes the MAIN program for KONPACT-1 on file COMPIL. The flow chart is given in Figure 119 and the program listing is given in Figure 120.

```
      PROGRAM PRECOM(INPUT,OUTPUT,COMP,TAPE5=INPUT,TAPE9=OUTPUT
     1,TAPE6=COMP)
C
C     ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C     PURPOSE - TO READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM
C     DIMENSIONS AND SET UP THE MAIN PROGRAM FOR KONPACT-1 PROGRAMS
C     DATE WRITTEN - DECEMBER 1975
C
      DIMENSION CARD(20)
      DATA HNXME,HNRME,HNUME,HNYME/4HNXM=,4HNRM=,4HNUM=,4HNYM=/
      DATA HMSBE,HMTBE,HCH,HK8BB/4HMSB=,4HMTB=,2HC ,4HK   /
      DATA HK1BB,HK2BB,HK3BB,HK4BB/4HK1  ,4HK2  ,4HK3  ,4HK4  /
      MS1F=0 $ MS2F=0 $ MS3F=0 $ MS4F=0 $ MS5F=0
C
C     INITIALIZE MAXIMUM SYSTEM DIMENSIONS
C
      NXM=0 $ NRM=0 $ NUM=0 $ NYM=0 $ MSB=0 $ MTB=0
C
C     READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM DIMENSIONS
C
  100 CONTINUE
      READ(5,120)CARD
  120 FORMAT(20A4)
      IF(EOF(5))220,140
  140 CONTINUE
      DECODE(4,160,CARD(1))CC,DUMMY
  160 FORMAT(A2,A2)
      IF(CC.EQ.HCH)GO TO 100
C
C     SET THE PROGRAM FLAGS
C
      CODE=CARD(2)
      IF(CODE.EQ.HK1BB)MS1F=1
      IF(CODE.EQ.HK1BB)GO TO 100
      IF(CODE.EQ.HK2BB)MS2F=1
      IF(CODE.EQ.HK2BB)GO TO 100
      IF(CODE.EQ.HK3BB)MS3F=1
      IF(CODE.EQ.HK3BB)GO TO 100
      IF(CODE.EQ.HK4BB)MS4F=1
      IF(CODE.EQ.HK4BB)GO TO 100
      IF(CODE.EQ.HK8BB)MS5F=1
      IF(CODE.EQ.HK8BB)GO TO 100
C
C     SET THE MAXIMUM SYSTEM DIMENSIONS
C
      CODE=CARD(1)
      DECODE(4,180,CARD(2))MAX,DUMMY
  180 FORMAT(I3,A1)
      IF(CODE.EQ.HNXME)NXM=MAX
      IF(CODE.EQ.HNXME)GO TO 100
      IF(CODE.EQ.HNRME)NRM=MAX
      IF(CODE.EQ.HNRME)GO TO 100
      IF(CODE.EQ.HNUME)NUM=MAX
      IF(CODE.EQ.HNUME)GO TO 100
      IF(CODE.EQ.HNYME)NYM=MAX
      IF(CODE.EQ.HNYME)GO TO 100
      IF(CODE.EQ.HMSBE)MSB=MAX
      IF(CODE.EQ.HMSBE)GO TO 100
      IF(CODE.EQ.HMTBE)MTB=MAX
      IF(CODE.EQ.HMTBE)GO TO 100
C
C     IF DATA CARD IS IN ERROR PRINT ERROR MESSEGE
C
```

Figure 120. Program PRECOM Program Listing

```
      WRITE(9,200)CARD
  200 FORMAT(1H1,//,1X,25HERROR IN PRECOMPILER DATA,//,1X
     1,18HLAST CARD READ =AS,//,1X,20A4)
      STOP 111
C
C     CALCULATE DIMENSIONS WHICH ARE USEFUL TO COMPUTE
C     MAXIMUM SCRATCH ARRAY DIMENSIONS REQUIRED
C
  220 CONTINUE
      NXRM=NXM+NRM
      NXUM=NXM+NUM
      NYUM=NYM+NUM
      NRUM=NRM+NUM
      NXRUM=NXM+NRUM
      NXRYM=NXRM+NYM
      MWORD=17 $ NRSM=1
      NDM11=MAX0(MWORD,NXM,NRM,NRSM)
      NDM12=MAX0(NXUM,NRM)
      NDM21=MAX0(NRM,NXM,NRSM)
      NDM22=MAX0(NXM,NUM)
      MM=MAX0(NUM,NRM)
C
C     CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S1
C     TO USE THE VARIOUS KONPACT-1 PROGRAMS
C
      MS111=1+3*NXM+2*(NYUM)+NXRYM*(2*NXM+NYUM)+NRM
      MS112=1+MSB*(14*NXRUM+3)
      MS11=MAX0(MS111,MS112)
      MS121=MS111+MTB*15
      MS122=MS112
      MS12=MAX0(MS121,MS122)
      MS131=MS111+MSB*(3+NXRUM+NXM)+NRM*(MSB+1)
      MS132=MS112
      MS13=MAX0(MS131,MS132)
      MS14=MS11
      MS15=1+NXM*(NRUM+NRM)+NRM*(2*NRM+3*NUM)+3*NXRUM
     1+NDM11*NDM12+NDM21*NDM22+NUM
C
C     CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S2
C     TO USE THE VARIOUS KONPACT-1 PROGRAMS
C
      MS211=1+NXRM*NXUM
      MS212=10000
      MS21=MAX0(MS211,MS212)
      MS221=MS211
      MS222=13+NRM*NUM+MTB*(48+MTB+NRUM)
      MS22=MAX0(MS221,MS222)
      MS231=MS211
      MS232=1+NRM*NUM+MSB*MM*(MSB*MM+NRUM)+MSB*NXRM*NXUM
      MS23=MAX0(MS231,MS232)
      MS24=MS211
      MS25=1+(NXRM+NRM)*NXUM
C
C     CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S3
C     TO USE THE VARIOUS KONPACT-1 PROGRAMS
C
      MS31=1+17*NXRUM
      MS32=MS31
      MS331=MS31
      MS332=1+4*MSB+MM*(2*MM+NRUM)*MSB
      MS33=MAX0(MS331,MS332)
      MS34=MS31
      MS35=2*MS31
C
C     IF NO SPECIFIC PROGRAMS ARE READ SET ALL PROGRAM FLAGS TO 1
C
```

Figure 120. Program PRECOM Program Listing (Continued)

```
      IF((MS1F.NE.0).OR.(MS2F.NE.0).OR.(MS3F.NE.0).OR.(MS4F.NE.0).OR.
     1(MS5F.NE.0))GO TO 221
      MS1F=1 $ MS2F=1 $ MS3F=1 $ MS4F=1 $ MS5F=1
C
C     CALCULATE MAXIMUM SCRATCH ARRAY DIMENSIONS NEEDED
C
  221 CONTINUE
      IF(MS1F.EQ.1)GO TO 222
      MS11=1 $ MS21=1 $ MS31=1
  222 CONTINUE
      IF(MS2F.EQ.1)GO TO 224
      MS12=1 $ MS22=1 $ MS32=1
  224 CONTINUE
      IF(MS3F.EQ.1)GO TO 226
      MS13=1 $ MS23=1 $ MS33=1
  226 CONTINUE
      IF(MS4F.EQ.1)GO TO 228
      MS14=1 $ MS24=1 $ MS34=1
  228 CONTINUE
      IF(MS5F.EQ.1)GO TO 230
      MS15=1 $ MS25=1 $ MS35=1
  230 CONTINUE
      MS1=MAX0(MS11,MS12,MS13,MS14,MS15)
      MS2=MAX0(MS21,MS22,MS23,MS24,MS25)
      MS3=MAX0(MS31,MS32,MS33,MS34,MS35)
      MS4=1
C
C     COMPUTE MEMORY REQUIRED FOR SCRATCH ARRAYS
C
      MST=MS1+MS2+MS3+MS4
C
C     SET THE MEMORY REQUIRED FOR THE PROGRAM CODE
C
      MPT=30000
C
C     COMPUTE TOTAL MEMORY REQUIRED TO EXECUTE KONPACT-1 PROGRAM AND
C     PRINT THE FIELD LENGTH REQUIRED IN OCTAL BASE
C
      MT=MST+MPT
      WRITE(9,240)
  240 FORMAT(////)
      WRITE(9,260)
  260 FORMAT(/,10X,56(1H*))
      WRITE(9,280)MT
  280 FORMAT(/,10X,50HFIELD LENGTH REQUIRED FOR EXECUTING KONPACT-1 =
     1,O6)
      WRITE(9,260)
C
C     WRITE MAIN PROGRAM FOR KONPACT-1 PROGRAM ON COMP FILE
C
      WRITE(6,300)
  300 FORMAT(38H      PROGRAM MAIN(BINPUT,INPUT,NDATA,
     1,40HUDATA,OUTPUT,TAPE5=BINPUT,
     2,/,36H     1TAPE6=INPUT,TAPE7=NDATA,TAPE8=
     3,40HUDATA,TAPE9=OUTPUT,VDATA,
     4,/,38H     2TAPE4=VDATA,SCRTCH,TAPE3=SCRTCH))
      WRITE(6,320)
  320 FORMAT(39HC     ANALYSIS - A F KONAR / J K MAHESH
     1,40H - THE HONEYWELL INC
     2,/,44HC     PURPOSE - TO SET UP MAXIMUM DIMENSIONS)
      WRITE(6,330)
  330 FORMAT(34H      COMMON /DIM/ MS1,MS2,MS3,MS4
     1,26H,MAXN,MAXM,NXM,NRM,NUM,NYM,/
     2,42H     1,MM,MP,MQ,MR,MSB,NR,MS,MN,MTR,MST,MT)
      WRITE(6,340)MS1,MS2,MS3,MS4
  340 FORMAT(22H      COMMON /SC1/ S1(,I5,1H),/,
```

Figure 120. Program PRECOM Program Listing (Continued)

```
    1         22H       COMMON /SC2/ S2(,I5,1H),/,
    2         22H       COMMON /SC3/ S3(,I5,1H),/,
    3         22H       COMMON /SC4/ S4(,I5,1H))
      WRITE(6,360)
  360 FORMAT(38HC       MAXIMUM SCRATCH ARRAY DIMENSIONS)
      WRITE(6,380)MS1,MS2,MS3,MS4
  380 FORMAT(6X,4HMS1= ,I5,7H $ MS2= ,I5,7H $ MS3= ,I5,7H $ MS4= ,I5)
      WRITE(6,400)
  400 FORMAT(31HC       MAXIMUM SYSTEM DIMENSIONS)
      WRITE(6,420)NXM,NPM,NUM,NYM,MSH,MTB
  420 FORMAT(6X,4HNKM= ,I3,7H $ NRM= ,I3,7H $ NUM= ,I3,7H $ NYM=
     1,I3,7H $ MSH= ,I3,7H $ MTR= ,I3)
      WRITE(6,440)
  440 FORMAT(40HC       CALL KONPACT ORGANIZING SUBROUTINE)
      WRITE(6,460)
  460 FORMAT(16H       CALL KORG1,/,25HC       STOP EXECUTION
     1,/,10H       STOP,/,9H       END)
      STOP
      END
```

Figure 120. Program PRECOM Program Listing (Concluded)

REFERENCES

1. Hink, G. R., et. al., "A Method for Predicting the Stability Characteristics of Control Configured Vehicles," Volume II - FLEXSTAB 2.01.00 User's Manual, AFFDL-TR-74-91, November 1974.

2. VanDierendonck, A. J., Hartmann, G. L., "Quadratic Methodology," Volume II - Documentation for Computer Program, Honeywell Report No. F0161-FR, October 1973.

3. "Scope Reference Manual," for the Control Data CYBER 70/Models 72, 73, and 74 and 6000 series computers, publication number 60307200, 1973.

4. Phillips, A. C., Rosen, R. W., "A Note on Dynamic Data Storage in FORTRAN IV," Computer Journals, Vol. 18, No. 4, November 1975.

5. Konar, A. F., "Improved Iterative Algorithm for Solving the Lyapunov Equation," Research Memo, MR-10702, Honeywell Systems and Research Center, Minneapolis, Minnesota, June 1968.

6. Stein, A., "Practicalization Algorithms," Research Memo, MR-10861, Honeywell Systems and Research Center, Minneapolis, Minnesota, January 1970.

☆ U. S. GOVERNMENT PRINTING OFFICE: 1976 — 657-630/1058